This is to certify that the

thesis entitled

AUTOMATIC IDENTITY VERIFICATION USING
FACES, FINGERPRINTS, SPEECH

presented by

Yatin Subhashchandra Kulkarni

has been accepted towards fulfillment
of the requirements for

Master's degree in Computer Science
& Engineering

Major professor

Date Dec 21, 1998

Automatic Identity Verification using Faces, Fingerprints and Speech

By

Yatin S. Kulkarni

A Thesis

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Computer Science and Engineering

1998

Abstract

Automatic Identity Verification using Faces, Fingerprints and Speech

By

Yatin S. Kulkarni

In the age of Internet communication and commerce, ensuring the security of transactions is a major concern. To ensure the total security of the billions of transactions taking place everyday, it is necessary to prevent the information being exchanged from being intercepted during transmission and to establish the identity of the sender and/or the receiver. While increasingly powerful and sophisticated data encryption algorithms are being made available to avoid the interception of information, age-old techniques of user-name and password are still the predominant means of establishing the identity of the sender and/or the receiver. Biometrics is a technique that is capable of establishing an individual's identity with a high degree of confidence on the basis of his/her physiological and/or behavioral characteristic. Of the various biometric techniques available [1], we are mainly interested in *faces*, *fingerprints*, and *speech*. Recognition by face is probably the most common method that humans use to establish a person's identity. Fingerprint verification has proved to be the most reliable way of verifying the identity of a person and is by far the most

mature biometric technique. Speaker verification, wherein a person's voice patterns are used to authenticate his/her identity claim, has been shown to be readily accepted and widely deployed biometric technique. In this thesis we have designed and implemented an automatic identity verification system that authenticates an individual's identity claim by integrating the above three biometric techniques through a robust decision fusion scheme. Our goal is to improve the performance of the integrated system, in terms of false accept and false reject rates, beyond those possible by using each of the techniques separately, while still meeting the response time requirements so as to make a real time implementation possible.

To My Wife, Kaushika

Acknowledgments

This thesis could not have been completed but for the time and patience of the numerous people who generously allowed me to collect face, fingerprint, and speech samples for use in my research. I sincerely thank each one of them for their cooperation. Special thanks are due to the members of the Pattern Recognition and Images Processing laboratory of the Department of Computer Science and Engineering for continually accommodating my requests for more and more data in their busy schedules and for giving me a number of useful insights into my research. Above all, I wish to express my gratitude to Professor Anil K. Jain, my thesis advisor, for his generous support, guidance and patience throughout my research. His attention to detail has inspired me time and again to critically analyze and improve my work. I have benefited greatly from his vast experience and valuable suggestions that he has shared with me over the course of this work. I will be forever indebted to him for providing me this wonderful opportunity to work in this field.

I would like to thank Profs. John Weng and Jack Deller for serving on my thesis committee. I am very grateful to Dr. Weng for his helpful suggestions and constructive criticism that contributed towards improving the quality of my research. Special

thanks are due to Dr. Deller for sharing his knowledge about speech recognition using hidden Markov models that contributed significantly to Chapter 4 of this thesis. I would also like to express my thanks to Prof. George Stockman for always keeping an open door for me to discuss research with him at any time throughout my work. I also want to thank all the faculty members from the Departments of Computer Science and Engineering, Electrical and Computer Engineering, and Statistics and Probability for the excellent instruction that I received during the course of my Masters degree.

I also wish to express my sincere gratitude to Dr. Lin Hong for his help and guidance on every aspect of my research and also for sharing his fingerprint data and source code. I would also like to thank Aditya Vailaya, Scott Connell, Nicolae Duta, Vera Bakic and my other PRIP colleagues for helping me throughout my work. Special thanks are also due to Stefaan Delcroix for helping me academically as well as for providing moral support and fun times outside school.

I am grateful to the Department of Computer Science and Engineering for its generous financial support without which my research would not have been possible. I also wish to thank the system administrators in the Department for maintaining excellent computing facilities and satisfying my ever increasing disk space requirement. I also appreciate the enthusiastic helpfulness of the Departmental support staff, particularly Linda Moore, Cathy Davison, and Mary Gebbia at all times.

Special thanks to my parents and my brother, Ketan, who stood by me in troubled times and made all this possible. My mother's undying love and faith enabled me to overcome a number of obstacles and achieve my goals. Special thanks are also due

to my

for th

my w

encou

in the

gratit

to my father-in-law Dr. S. V. Shanbhag and my mother-in-law Mrs. S. S. Shanbhag for their moral support over the past six years. Finally, I am most thankful to my wife, Kaushika, for her love and support over the last six years. Her constant encouragement helped me tremendously during my studies. She also was a great help in the data collection process. I dedicate this thesis to her with all my love and sincere gratitude.

LIST

LIST

1 In
1.1
1.2
1.3
1.4
1.5
1.5.1
1.5.2
1.6
1.7
1.8
1.9
1.10
1.11

2 Fa
2.1
2.2
2.2.1
2.2.2
2.3
2.4

3 F
3.1
3.2
3.2.1
3.3
3.3.1
3.4
3.5

Table of Contents

LIST OF TABLES x

LIST OF FIGURES xi

1 Introduction 1
1.1 An Ideal Personal Identity Verification System 2
1.2 Biometrics 3
1.3 Requirements of a biometric characteristic 4
1.4 Biometric Technologies Prevalent Today 5
1.5 A Biometric based Personal Identity Authentication System 8
1.5.1 Operational Mode 9
1.5.2 Accuracy 10
1.6 Combination of multiple biometrics 11
1.7 Problem Statement 14
1.8 Intended use and Constraints 14
1.9 Contributions 14
1.10 Overview of the thesis 15
1.11 Summary 15

2 Face Verification 17
2.1 Face Detection 19
2.2 Face Recognition 21
2.2.1 Training Stage 24
2.2.2 Operational Stage 26
2.3 Experimental Results in Literature 27
2.4 Summary 28

3 Fingerprint Verification 29
3.1 Fingerprint Image Acquisition 29
3.2 Feature Extraction 32
3.2.1 Minutiae Extraction 34
3.3 Fingerprint Matching 35
3.3.1 Minutiae Matching 36
3.4 Experimental Results in Literature 40
3.5 Summary 40

4 Sp
4.1
4.1.1
4.1.2
4.1.3
4.2
4.3
4.4
4.5

5 D
5.1
5.2
5.3
5.4

6 I
6.1
6.2
6.3
6.4
6.5
6.6

7 I
7.1
7.2
7.3
7.3.
7.3.
7.3.
7.4

8
8.1
8.2

4 Speaker Verification **43**
4.1 Feature Extraction 45
4.1.1 Speech Acquisition and Preprocessing 51
4.1.2 Speech Segmentation 52
4.1.3 Cepstral Analysis 58
4.2 Speaker Modeling 62
4.3 Pattern Matching 73
4.4 Experimental Results in Literature 76
4.5 Summary 77

5 Decision Fusion **79**
5.1 Formulation 81
5.2 The Neyman-Pearson Rule 84
5.3 Linear Discriminant Functions 86
5.4 Summary 87

6 Integrated Biometric System **88**
6.1 Data Acquisition Module 90
6.2 Enrollment Module 90
6.3 Template Database 92
6.4 Verification Module 93
6.5 Commercially Available Biometric Systems 94
6.6 Summary 96

7 Experimental Results **97**
7.1 Database 99
7.2 System Training 103
7.3 Performance Evaluation 107
7.3.1 Performance of the Neyman-Pearson decision fusion schema 107
7.3.2 Performance of the linear discriminant function decision fusion schema 108
7.3.3 Verification Speed 108
7.4 Summary 109

8 Summary and Future Research **111**
8.1 Summary 111
8.2 Future Research 113

1.1

7.1
7.2

List of Tables

1.1 Comparison of various Biometric technologies [2]. 7

7.1 Various factors affecting the performance of different biometric systems. . 98
7.2 Wall-clock times for the various verification systems. 109

List of Figures

1.1 Examples of nine different biometric technologies. 6
1.2 A generic biometric identity verification system. 9

2.1 The architecture of a face-based verification system. 18
2.2 A typical facial image obtained using our imaging setup. 19
2.3 Our face image segmentation algorithm. 21
2.4 The first nine eigenfaces. 24

3.1 Architecture of the fingerprint verification system. 30
3.2 Different kinds of fingerprints: (a) inked, (b) latent, and (c) live-scan. . . 31
3.3 The live-scan fingerprint scanner from Digital Biometrics. 32
3.4 Examples of minutiae and their characterization. 33
3.5 Flowchart of the minutiae extraction algorithm after Jain *et al.* [2]. . . . 35
3.6 Fingerprint matching problem: (a) and (b) are two different impressions from the same finger. 37
3.7 Flowchart of the minutiae matching algorithm. 41

4.1 A block diagram of the HMM-based Speaker Verification Module. 45
4.2 A schematic diagram of the human speech production mechanism [3]. . . 46
4.3 A discrete-time model representing the human speech production system [3]. 48
4.4 Vocal-tract models for different sounds [3]. 49
4.5 Vocal-tract magnitude spectrum for letter 'a' for two different speakers. . 50
4.6 Plots of the log energy for speech acquired under different environments. 53
4.7 The flowchart of the algorithm for word boundary detection. 55
4.8 Output of the speech segmentation algorithm. 58
4.9 A fully connected three-state hidden Markov model. 63
4.10 A left-to-right five state Hidden Markov Model. 64

5.1 Integration of different biometric characteristics (after Jain *et al.* [4]). . . 80

6.1 The block diagram of the integrated system for personal verification. . . 89
6.2 GUIs for data acquisition during enrollment. 91
6.3 User Database: (a) The users enrolled in the system are shown in the list, (b) For each user, three type of data are defined. 92
6.4 GUIs for the enrollment module. 93
6.5 GUIs for the verification module. 94
6.6 The result of a successful verification. 95

7.1
7.2
7.3
7.4
7.5
7.6
7.7
7.8
7.9
8.1

7.1 An ideal receiver operating characteristics curve. 99
7.2 A typical example of the nine face images acquired for each user. 100
7.3 A typical example of the ten fingerprint images of good quality acquired for each user. 101
7.4 The waveform of a typical speech sample in our database. 102
7.5 The genuine and impostor distributions for face matching. 104
7.6 The genuine and impostor distributions for fingerprint matching. 105
7.7 The genuine and impostor distributions for the speaker verification system. 106
7.8 The receiver operating characteristics for the Neyman-Pearson rule. . . . 108
7.9 The receiver operating characteristics for the Linear Discriminant function. 109

8.1 Example of the temporal variation present between training and test face images. 114

Chapter 1

Introduction

Today, the Internet is changing the very nature of business. More and more routine business transactions such as exchange of important documents, transfer of funds, etc. are being conducted electronically. While the companies, institutions, and the people involved in these transaction stand to benefit from the speed and efficiency that the Internet has to offer, it has also made it possible for hi-tech computer criminals to gain access to sensitive information and monetary resources. Hence, with the growing volume of Internet commerce, there is a growing need for ensuring the security of the billions of electronic transactions taking place.

To ensure fool proof security, it is necessary to establish the identity of the receiver and/or the sender and to ensure that the information is not intercepted during transit. Currently, it is possible to ensure that the information is not intercepted during transit by means of powerful and sophisticated data encryption programs. However, the age-old method of requesting a user-name and a password is still the only means of establishing the identity of the sender and/or the receiver.

Since the user-name and password are something that one knows, this method of establishing a person's identity is called a *knowledge*-based method. The other traditional method, which is very popular in point-of-access control applications, relies on the user supplying a token such as a passport, a driver's license, etc., as proof of identity and is correspondingly called a *token*-based method.

Automatic personal identity authentication systems based on these two methods have been deployed on a large scale during the past few decades and have gained immense popularity. The most popular system being the combination of a magnetic stripe card and a *personal identification number* (PIN) that is still being used at millions of ATM machines world wide [1]. The popularity of these methods can be attributed to their simplicity, ease of use, the ability to give absolute yes/no answers and the ease of integrating them into the existing systems. However, their simplicity also makes them easily susceptible to fraud. Passwords and PINs can be stolen, misplaced or forgotten, and identity cards may be easily forged. Furthermore, since the proof of identity is something that one knows or one has, it is not possible to distinguish between an authorized user and an impostor who has fraudulently gained access to the relevant knowledge and/or token.

1.1 An Ideal Personal Identity Verification System

The pertinent question today then, is "Can we have an automatic personal identity verification system that is easy to use, 100% accurate and impossible to circumvent?" The answer to this question lies in understanding how the three requirements can

be satisfied. By "easy to use" we mean that the user is not required to memorize a cumbersome password or a personal identification number and neither is he required to carry around a magnetic stripe card or any other proof of identity. By "100% accurate" we imply that the system never allows an impostor to gain access and neither does it ever reject a genuine individual. And finally, by "impossible to circumvent" we mean that an impostor can by no means fool the system into recognizing him/her as a genuine individual.

Unfortunately such a system is still, at best, an elusive goal.

1.2 Biometrics

As humans, we identify a person by his/her face and/or voice. When a more reliable proof of identity is required, use of fingerprints is preferred. Thus, we rely on the variations in various physiological and behavioral characteristics amongst different people for identification. Biometrics is a collection of technologies that attempts to measure these variations so as to make automatic personal identification, using physiological and behavioral characteristics, possible. It differs from *knowledge-based* and *token-based* personal identification systems, in that, it makes use of something that *one is* or *one does* to identify that individual. Since it is impossible to steal a physiological characteristic from an individual and difficult to imitate an individual's behavioral characteristic, biometrics is inherently better suited for automatic personal identification and/or identity authentication.

1.3 Requirements of a Biometric Characteristic

Based on the definition of biometrics given above, it is easy to see that not all human physiological and behavioral characteristics can be used as biometrics. The following requirements must be satisfied by a physiological and behavioral characteristic to be useful as a biometric [2]: (*i*) *universality*, which means that every individual must posses that characteristic, (*ii*) *uniqueness*, which implies that no two individuals should be identical with respect to that characteristic, (*iii*) *permanence*, which states that the characteristic must not change beyond reasonable limits over a period of time and neither must it be easily changeable by artificial means, and (*iv*) *collectability*, which requires that the characteristic must be suitable for being measured quantitatively. Additional constraints are imposed on the characteristics due to the need for a biometric system to be of practical use. These constraints are: (*i*) *performance*, which refers to the resources required for constructing the biometric system, the time required for the system to measure the biometric, process it and make a decision, the identification accuracy that can be achieved using that biometric, and the robustness of the system to variations in operating environment, (*ii*) *acceptability*, which indicates the ease of use and the willingness of the intended users towards using it on a regular basis, and (*iii*) *circumvention*, which indicates the possibility of the system being fooled by an impostor attempting to gain access by fraudulent means.

1.4 Biometric Technologies Prevalent Today

To date, nine biometrics namely *face*, *fingerprints*, *speech*, *iris*, *retinal blood vessels pattern*, *hand geometry*, *hand vein pattern*, *facial thermograms*, and *signature* have proved to be quite successful and a number of practical systems have also been built using these biometrics. Face, fingerprints, iris, retinal patterns, hand geometry, hand vein patterns, and facial thermograms are *physiological* characteristics whereas signature is a *behavioral* characteristic. Speech is a physiological as well as behavioral characteristic. Typically, biometrics that are of a physiological nature are preferred over those of a behavioral nature, since they exhibit smaller *intra-class* variations. Furthermore, systems based on physiological biometrics are more difficult to circumvent than those based on behavioral biometrics. Figure 1.1 depicts examples of each of the above nine biometrics.

Each of the nine biometrics, described above, satisfy the seven requirements stated in Section 1.3 to varying degrees. Table 1.1 shows a brief comparison of the nine biometric techniques along the seven requirements [2]. Depending upon the intended application, the usefulness of a specific biometric technique varies. No single biometric technique can outperform all the other techniques in all application domains. For example, although fingerprint and iris-based biometric techniques perform much better than speech-based systems, in terms of accuracy, in telephony based applications, speaker verification is the most economical choice.

Besides the techniques mentioned above, a number of other biometric techniques have been investigated or are currently under study, which include *ear shape*, *gait*,

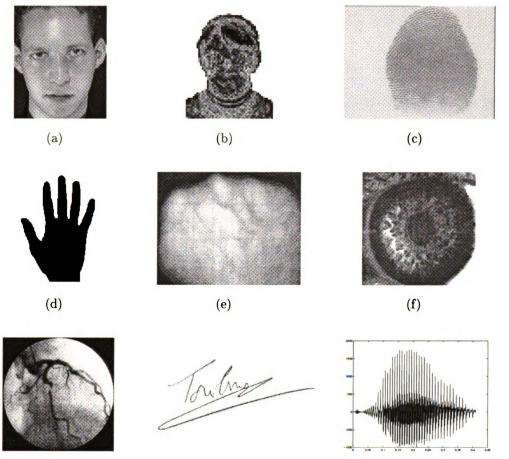


Figure 1.1: Examples of nine different biometric technologies.

keystroke dynamics, *body odor*, *acoustic emission of writing lip shape*, *DNA*, *etc.* Although each of these techniques has its own advantages, so far, none of them can achieve an accuracy that is comparable to the nine different techniques mentioned above or can be implemented fully automatically. In fact, they do not have a strong potential to become a valid biometric technique to be used widely in the near future.

Biometrics	Universality	Uniqueness	Permanence	Collectability	Performance	Acceptability	Circumvention
Face	high	low	medium	high	low	high	low
F. Thermograms	high	high	low	high	medium	high	high
Fingerprint	medium	high	high	medium	high	medium	high
Hand Geometry	medium	medium	medium	high	medium	medium	medium
Hand Vein	medium	medium	medium	medium	medium	medium	high
Iris	high	high	high	medium	high	low	high
Retinal Scan	high	high	medium	low	high	low	high
Signature	low	low	low	high	low	high	low
Speech	medium	low	low	medium	low	high	low

Table 1.1: Comparison of various Biometric technologies [2].

1.5 A Biometric based Personal Identity Authentication System

Having studied the various biometric technologies, let us now address the issue of designing an automatic personal identity authentication system. A generic automatic personal identity authentication system based on biometrics is depicted in Figure 1.2. Clearly, before an individual can use the system, he/she must be *enrolled in* the system. This is similar, for example, to going to a bank and opening a bank account so that you may receive an ATM card, which would then allow you to withdraw money anytime at an ATM machine. Consequently, any biometric system must consist of at-least two modules: (*i*) *enrollment* module and (*ii*) *verification* module. From an engineering perspective, the enrollment module must be able to (*i*) obtain a raw digital representation of the desired biometric, (*ii*) process the raw digital representation by means of a feature extractor so as to generate a compact but expressive representation, referred to as a *template*, and (*iii*) store the template in a central database or a magnetic stripe card or a smart card to be issued to the user. The verification module is to be used at the point-of-access and it should be able to capture the same biometric characteristic as the enrollment module, convert it into the same format as the template and finally compare the two so as to verify the identity claimed by the individual.

Apart from these two essential modules, a practical automatic personal identity verification system based on biometrics may also incorporate auxiliary modules such as a module for database management (in case a central database is being used),

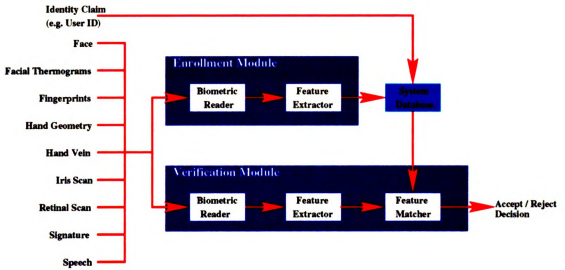


Figure 1.2: A generic biometric identity verification system.

modules for doing off-line training, *etc.* Such modules may not be directly related to the biometric technology being used, but are required nevertheless to make a biometric system practical to use on a daily basis.

1.5.1 Operational Mode

There are two possible ways in which the identity of an individual can be established: (*i*) *verification*, wherein the individual submits a claim of identity and the system verifies the validity of the claim, and (*ii*) *identification*, wherein the individual does not make any prior claim of identity and the system assigns an identity to the individual. In the verification mode, the system makes an one-to-one comparison between the input representation and the template corresponding to the claimed identity and makes a decision as to accept or reject the claim. In the identification mode, the system makes one-to-many comparisons between the input representation and the entire

template database. A measure of similarity or dissimilarity is computed for each comparison made and then the individual is assigned the identity of the template that resulted in the highest (lowest) value of the similarity (dissimilarity) measure.

1.5.2 Accuracy

Given a biometric system that is designed to operate either in the verification mode or the identification mode, we can divide the entire population of probable users into two sets: (*i*) *genuine individuals*, users enrolled in the system and authorized to access the resources being protected by the system, and (*ii*) *impostors*, users not enrolled in the system and not authorized to access the resources being protected by the system, but wanting to do so by fraudulent means. The biometric system either accepts or identifies a user as a genuine individual or rejects a user as an impostor. Consequently, the following four situations may occur: (*i*) a genuine individual is accepted, (*ii*) a genuine individual is rejected, (*iii*) an impostor is accepted, and (*iv*) an impostor is rejected. Clearly, situations (*i*) and (*iv*) are correct system responses whereas situations (*ii*) and (*iii*) are erroneous system responses. It is interesting to note that situations (*ii*) and (*iii*) never occur in a knowledge-based or token-based identification system. A PIN or a password is either correct or incorrect. However, all of the biometric characteristics studied above exhibit a significant amount of intra-class variations. Due to these variations, a biometric system cannot make an absolute *yes* or *no* decision, but rather it associates a *confidence level* with the decision, which may be represented as a probability value. Therefore, the performance of a biometric

system is measured in terms of two error rates, (*i*) *false acceptance rate* (FAR), the probability that an impostor is accepted as a genuine individual, and (*ii*) *false reject rate* (FRR), the probability that a genuine individual is rejected as an impostor. These two error rates can be evaluated empirically for a given biometric system by generating a genuine distribution and an impostor distribution of the similarity (dissimilarity) measure used. It is easy to see that FAR and FRR are dual of each other. Ideally, we would like both of them to be zero so that an impostor is never accepted and a genuine individual is never rejected. However in practical systems, a low FAR results in a high FRR and *vice versa*. Depending upon the intended domain of application, it may be desirable to either have a low FAR and a tolerable FRR, or, have a low FRR and an admissible FAR.

In some cases the performance of a biometric system is also measured in terms of the *authentic acceptance rate*, which is given as $(1 - FAR)$, and the *equal error rate*, which is defined as the error rate for which $(FAR = FRR)$.

1.6 Combination of multiple biometrics

The individual use of different biometric characteristics result in systems that vary widely in terms of FAR and FRR. At one extreme we have biometric characteristics such as face and dynamic signature that result in a system with low FRR but a high FAR, while at the other extreme, systems based on fingerprints, retinal scans and iris have a very low FAR but an unacceptably high FRR. The primary goal of combining multiple biometrics is, then, to improve the FAR and the FRR of the integrated

system while meeting the response time requirement of a real time system. A number of researchers have proposed automatic personal identification and/or verification systems that combine two or more different biometrics.

Brunelli and Falavigna [5] have proposed a system that integrates face recognition and speaker recognition. They have decomposed the face recognition and speaker recognition subsystems into three and two single feature classifiers, respectively. Various methods of combining the scores resulting from these five classifiers are then studied. They report a authentic acceptance rate of 98% for the integrated system as opposed to the 88% and 91% authentic acceptance rates provided by the speaker and face recognition systems, respectively.

Dieckmann *et al.* [6] have proposed the personal authentication system, SESAM, that integrates three different biometric cues from two different data sources: one static cue derived from an image of the face and two dynamic cues, the spectrum of the sound and the lip motion of a person saying his/her name in front of the system. The results of the individual classifiers are combined using a 2-from-3 approach. They report a FRR of 6.6% and FAR of 0.4% for the integrated identification system as compared to a FRR of 19.7%, 12.5% and 18.2% and a FAR of 2.1%, 1.3% and 7.2% for speech, lip movement and face image systems, respectively.

Duc *et al.* [7] also propose a system that integrates speech-based authentication and face-based authentication. However, they model the joint distributions of the speech and face matching scores for genuine individuals and impostors as Gaussian distributions and then use a maximum a posteriori (MAP) classifier to make the accept/reject decision. They report a FAR of 0.54% and FRR of 0.0% for the integrated

system as opposed to a FAR of 3.6% and 6.7% and a FRR of 7.4% and 0.0% for the face and the speech systems, respectively.

Hong and Jain [8] have proposed an automatic personal identification system that integrates faces and fingerprints. First, the impostor distributions for faces and fingerprints are estimated from empirical data. The system then attempts to minimize FRR while holding FAR constant at a desired value. This is achieved by varying the decision threshold for the fingerprint system in accordance with the face matching score and the desired FAR. For a FAR of 0.001%, they report a FRR of 9.8% for the integrated system as opposed to a FRR of 64.1% for the face system and a FRR of 14.9% for the fingerprint system.

The gain in performance reported in all the above cases is impressive enough to warrant further research into this topic. We have chosen to integrate face, fingerprint, and speech for the following reasons:

1. *Popularity*: Face, fingerprint, and speech are the most popular and widely accepted biometrics today. These biometrics are accepted in a court of law as valid proofs of identity.

2. *Cost*: Speech is the most economical biometric and can be most easily integrated into several existing systems. The cost of the additional resources required for acquiring face and fingerprint is rapidly decreasing. Thus, amongst the nine biometrics discussed in Section 1.4, face, fingerprint, and speech offer the most cost efficient solution.

1.7 F

Our object
tion system
a decision f
what is pos

1.8 In

The resultin
establishmen
thousands. V
intended for u
ambient nois

1.9 Co

This thesis h
scheme that e
and (ii) the d
cation system
The face r
alone applica
platform and i

1.7 Problem Statement

Our objective in this thesis is to design and implement an automatic identity verification system that integrates face, fingerprint, and speaker verification modules through a decision fusion scheme with an objective of achieving a lower FAR and FRR than what is possible using these three individual biometrics.

1.8 Intended use and Constraints

The resulting automatic verification system is to be used in a small to medium sized establishment wherein the number of enrolled individuals is of the order of a few thousands. We further assume the users to be cooperative. Finally, the system is intended for use in a controlled environment, wherein illumination, background, and ambient noise are within specified limits.

1.9 Contributions

This thesis has resulted in the following contributions: (*i*) a robust decision fusion scheme that enables the integration of face, fingerprint, and speech for verification, and (*ii*) the design and implementation of a fully automatic personal identity verification system based on the proposed decision fusion scheme.

The face recognition and fingerprint verification modules were obtained as standalone applications for the Unix platform [8] and were ported to the Windows NT platform and integrated with a speaker verification module. The speaker verification

module was

Laboratori

by the auth

1.10

We begin

of fingerpr

Chapter

face, finge

of the int

Chapter

Chapter

topics in

1.11

Knowled

cure in

biometri

holds th

based o

that of

module was developed with the aid of the HTK toolkit [9] from Entropic Research Laboratories [10]. The system itself has evolved from a previously developed system by the author, called F2ID [11].

1.10 Overview of the thesis

We begin in Chapter 2 with a study of face verification techniques, followed by a study of fingerprint matching techniques in Chapter 3 and speaker verification techniques in Chapter 4. Then in Chapter 5 we address the decision fusion schemes that integrate face, fingerprint, and speech so as to improve the verification accuracy. The design of the integrated system and some implementation details are discussed at length in Chapter 6, followed by the experimental results in Chapter 7. Finally, we conclude in Chapter 8 by addressing the limitations of our system and outline the future research topics in this field.

1.11 Summary

Knowledge-based and token-based identity authentication systems are no longer secure in our modern electronic society. An identity authentication system based on biometrics overcomes the limitations of a knowledge-based or token-based system and holds the key to the security of various systems in the future. Furthermore, systems based on multiple biometrics are able to improve the accuracy of the system beyond that of a system based on a single biometric. Such multimodal biometric systems

are also more

fully automa

namely, face

fusion schem

based on th

are also more robust to fraud. The goal of this thesis is to design and implement a fully automatic personal identity authentication system that uses three biometrics, namely, face, fingerprint, and speech and demonstrate that our proposed decision fusion scheme improves the performance of the system beyond that of the systems based on the individual biometrics.

Chap

Face

Face, as a b

allows for fa

system can b

identificatio

based on fac

diagram of a

The face

video image

database of

the followin

the given in

and (*b*) ide

face against

Operate in a

Chapter 2

Face Verification

Face, as a biometric, has been primarily used in identification systems since its use allows for fast indexing into a large database. Its use as a biometric in a verification system can be considered as a logical extension of the principles underlying its use in identification system. Towards this end, we shall first discuss an identification system based on faces and then extend our discussion to face-based verification. The block diagram of a face-based verification system is depicted in Figure 2.1.

The face recognition problem can be formally stated as follows: "Given still or video images of a scene, identify one or more persons in the scene using a stored database of faces" [12]. In order to solve the above problem, it is necessary to solve the following two sub-problems: (*i*) detect the presence of one or more human faces in the given image or sequence of images, and segment the faces from the background, and (*ii*) identify the individuals, by matching a general unconstrained view of their face against a database of faces. The lack of a face recognition system, that can operate in an unconstrained environment, can be attributed to these two challenging

sub-problems. The problem of face detection and segmentation has received a considerable amount of attention [13, 14, 15, 16, 17, 18, 19, 20]. Although the performance of these systems in terms of detection rate and false alarm rate is acceptable, the detection time is far too high to be used in a real-time biometric system. Furthermore, face recognition from a general view remains, to this date, an open research problem. Hence, in the context of a real-time automatic personal identification system, face recognition is generally performed using static, controlled and well illuminated frontal facial images captured against a plain background. Such constraints, although unrealistic in a practical sense, allow for an efficient and fairly accurate solution to the face recognition problem. Let us now discuss the details of a face recognition system.

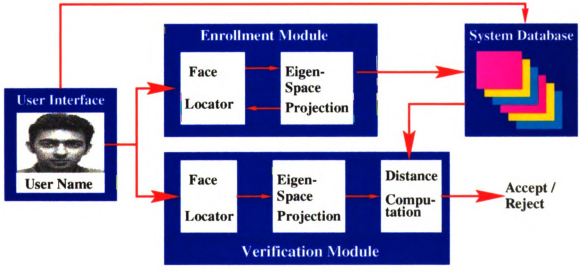


Figure 2.1: The architecture of a face-based verification system.

2.1

The ch
operati
laborat
2.2 sho
we are
the fac
steps:

1.

2.

2.1 Face Detection

The choice of a suitable face detection algorithm is dependent upon the intended operational environment. In our system, the facial images are obtained in a controlled laboratory environment under sufficient illumination and a plain background. Figure 2.2 shows a typical example of a facial image obtained using our imaging setup. Thus, we are able to employ a very simple edge-detection based algorithm in order to locate the face and segment it from the background. The algorithm consists of the following steps:

Figure 2.2: A typical facial image obtained using our imaging setup.

1. *Re-sampling*: The original image is 640x480 pixels in size. The size of the face portion in this image is approximately 320x360 pixels. The computational demand on the face recognition system for images of this size would render it incapable of operating in real time. Consequently, the original image is down-sampled to a size of 160x120 pixels. This results in a face sub-image of approximately 80x90 pixels.

2. *Smoothing*: The reduced image is smoothed using a Gaussian filter so as to retain the strong edges and eliminate the weak edges in the image.

3. *E*

to

re

is

im

be

ho

4. *Fa*

per

oth

emp

alo

the

occ

the

top

edg

com

righ

crop

mea

3. *Edge Detection*: Two edge operators, one to detect the vertical edges and one to detect the horizontal edges are then applied to the smoothed image and the resulting images are thresholded to obtain two binary edge images. The idea is to detect the left and the right extents of the face by using the vertical edge image and the top of the head using the horizontal edge image. The chin cannot be detected in this manner since it does not result in a significant edge in the horizontal edge image.

4. *Face detection and segmentation*: Two edge distributions (number of edge points per bin) are now computed: one along rows on the horizontal edge image and other along columns on the vertical edge image. The bin size is determined empirically for the given imaging setup (for our imaging setup bin size is 2 along rows as well as columns). Peaks in the two distributions are indicative of the presence of strong edges in the region spanned by the bins in which the peaks occur. To locate the left and the right bounds of the face, edge distribution for the vertical edge image is examined sequentially from left to right. Similarly, the top of the head is located by examining the edge distribution for the horizontal edge image. The location of the center of the face is then hypothesized by computing the centroid of all the edge pixels that lie within the left and the right bounds; an 80x90 window placed at the estimated center is then used to crop the face image. Finally, the cropped image is normalized to have a certain mean and standard deviation.

Figure

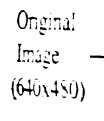
method is

the clothin

cooperative

feedback d

effectively



2.2 F

Face is a n

fact, it is

Consequen

standing

Figure 2.3 illustrates the various stages in our algorithm. Although this simple method is highly susceptible to artifacts in the image (such as vertical stripes on the clothing of the individual), it works rather well for our imaging setup and for cooperative users. In addition, we shall see in Chapter 6, that in the presence of feedback during the data acquisition process, the above method can be used very effectively for accurately extracting the face.

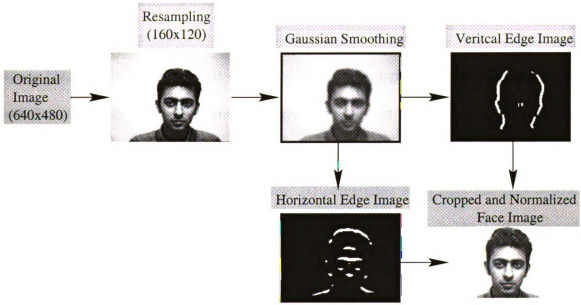


Figure 2.3: Our face image segmentation algorithm.

2.2 Face Recognition

Face is a natural choice of a physiological characteristic for use as a biometric and in fact, it is the most routinely used characteristic by humans to identify individuals. Consequently, a considerable amount of research has been devoted towards understanding the recognition process used by humans and machine vision systems that

require face image [21, 12, 22, 23, 24]. Numerous studies in psychophysics and neurophysiological literature have suggested that humans make use of facial features such as hair, eyes, nose, mouth, etc. to identify faces [12]. Based on these studies, early research in automatic face recognition systems focused on developing algorithms that attempted to measure the various facial features and represent a facial image based on these measurements [24]. However, these efforts met with limited success primarily due to the enormous computational complexity of the tasks involved. Popular techniques in face recognition now treat a facial image as a point in a high dimensional space [21, 23]. It is argued that the facial images belonging to a specific individual will form a set of points that are clustered in a compact region in this high dimensional space and clusters belonging to different individuals are well separated from each other. The recognition task involves: (*i*) computing a suitable distance metric between a given test image and a set of reference images in the high dimensional space and (*ii*) assigning the identity of the reference image with the smallest value of the chosen distance metric to the test image. Dimensionality reduction techniques such as principal component analysis (PCA), linear discriminant analysis (LDA), singular value decomposition (SVD), etc. reduce the dimensionality of the feature space so as to reduce the computational complexity of the problem while maintaining the inter-class separation [21, 23].

A recent study [22] of three well-known face recognition techniques, (*i*) *eigenface*, (*ii*) *elastic matching*, and (*iii*) *neural nets*, has shown by analysis and experimental results, that "the eigenface algorithm, which is essentially a minimum distance classifier, works well when lighting variation is small". Our decision to use the eigenface

appr

pers

grou

for ea

niques

appro

Fac

stages

eigenvec

respondi

eigenspac

the temp

onto the

the test *in*

tween this

neighbor d

The M

eigenvalu

approach

are *depic*

eigenface

approach for face recognition is based on the following reasons: (*i*) in the context of personal identification, control can be exerted over the lighting conditions, the background and the pose of the subject, (*ii*) a very compact template can be generated for each user by using the eigenface approach, (*iii*) efficient database retrieval techniques can be employed to retrieve the templates quickly [23], and (*iv*) the eigenface approach has been shown to be more accurate than the attribute-based approach [24].

Face recognition using the eigenface-based approach consists of the following two stages [21]: (*i*) *training stage*: given a set of input images, compute the eigenvalues and eigenvectors of the covariance matrix of these images; retain the M eigenvectors corresponding to the M highest eigenvalues; project each face onto this M-dimensional eigenspace so as to obtain an M-element feature vector for each face which then forms the template for that face, and (*ii*) *operational stage*: given a test image, project it onto the M-dimensional eigenspace, so as to obtain an M-element feature vector for the test image; compute the Euclidean distance, in the M-dimensional eigenspace, between this feature vector and each of the templates in the database; use the 1-nearest neighbor decision rule to establish the identity of the test image.

The M eigenvectors (represented as a 2D image) corresponding to the M highest eigenvalues resemble face images and hence the approach is termed as the *eigenface* approach. Examples of the first nine eigenfaces for a particular set of input images are depicted in Figure 2.4. We shall now discuss the mathematical details of the eigenface approach.

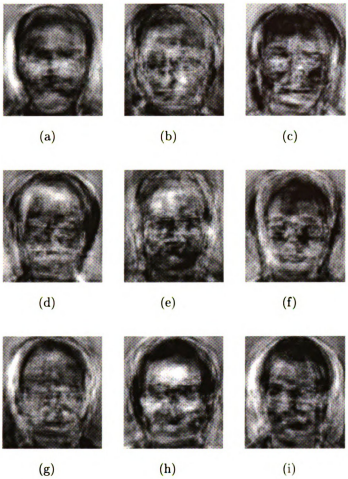


Figure 2.4: The first nine eigenfaces.

2.2.1 Training Stage

Given a face image, $I(x, y)$, as an $W \times H$ array of pixel intensities, it can be represented as a $W \times H$-dimensional feature vector, by concatenating the rows of $I(x, y)$. For our image size of 80×90 pixels, each face image maps to a point in the 7200-dimensional feature space. Due to the inherent similarity between different face images, the distribution of face images in this high dimensional space will not be random. Consequently, principal component analysis can be used to project this high dimensional

feature spac

Let us d

the number

is defined a

The co

where,

The di

eigenvalu

Furtherm

than 720

eigenvalu

computin

feature space to a lower dimensional subspace, also known as the eigenspace.

Let us denote the set of training face images as $\Gamma_1, \Gamma_2, \cdots, \Gamma_T$, where T denotes the number of images in the training set. The average face of the above set of images is defined as

$$\Psi = \frac{1}{T}\sum_{n=1}^{T}\Gamma_n.$$

The covariance matrix is then calculated as

$$C = \frac{1}{T}\sum_{n=1}^{T}\Psi_n\Psi_n',$$

where, $\Psi_n = \Gamma_n - \Psi$.

The dimension of this covariance matrix is 7200 by 7200 and calculating the eigenvalues and eigenvectors of a matrix of this size is "an intractable task"[21]. Furthermore, if the number of training images, $T \ll 7200$, there are only $T-1$ rather than 7200 meaningful eigenvectors ("The remaining eigenvectors will have associated eigenvalues of zero" [21]). The 7200 eigenvectors are, therefore, computed by first computing the eigenvectors of a $T \times T$ dimensional matrix, defined as

$$\Sigma = \frac{1}{T}\sum_{n=1}^{T}\Psi_n'\Psi_n.$$

Let v

est eige

as

The

vectors.

obtain t

2.2.2

Given a

matrix.

M-dim

Next,

Let $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_M$ denote the M eigenvectors of Σ, corresponding to the M highest eigenvalues. The M eigenvectors of the covariance matrix C are then obtained as

$$\mathbf{u}_l = \sum_{n=1}^{T} \mathbf{v}_{ln} \Psi_n, \qquad l = 1, \cdots, M.$$

The projection matrix is then given as $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_M)$, and the training vectors, $\Psi_1, \Psi_2, \cdots, \Psi_T$, are then projected onto the M-dimensional eigenspace to obtain the M-dimensional templates, $\Phi_1, \Phi_2, \cdots, \Phi_T$, as

$$\Phi_n = \mathbf{u}' \Psi_n, \qquad n = 1, \cdots, T.$$

2.2.2 Operational Stage

Given a set of T templates, $\Phi_1, \Phi_2, \cdots, \Phi_T$, the mean face, Ψ, and the projection matrix, $\mathbf{u}$, the operational stage takes a test face image, Γ, and projects it onto the M-dimensional eigenspace as

$$\Pi = \mathbf{u}'(\Gamma - \Psi).$$

Next, a 1-nearest neighbor classifier, operating in eigenspace, is used to establish

the identity of the test face image. The distance between the test pattern, Π, and a template, Φ_n, defined as $||\Phi_n - \Pi||$, where $|| \bullet ||$ denotes L_2 *norm*, is called Distance From Feature Space (DFFS) [21].

Identity verification using face can be performed by simply computing the minimum DFFS for the input image against the templates for the claimed identity, and comparing this DFFS against a pre-determined threshold. The threshold can be determined for a desired FAR or FRR by generating the impostor and genuine distributions for the DFFS for the given database of users.

2.3 Experimental Results in Literature

The eigenface method for identification has been tested by a number of researchers on several different databases [21, 12, 22]. It is well known that the method works best when the lighting and scale variations are small [21]. Consequently, for databases that satisfy the above requirement (e.g., the MIT database), the reported performance figures are as high as 97% correct classification, whereas for databases that do not satisfy these constraints (e.g., the Weizmann and Olivetti databases), the reported performance figures are around 80% correct classification. Zhang *et al.* [22] report a 66% correct classification rate for a database that was obtained by pooling together the MIT, the Olivetti, the Weizmann and the Bern databases.

2.4 S

Face recos
accepted
the most
practice.
works bes
against a
mented fr
if such co
more, ther
and that
accuracy.

Ideally
ence of a
the face f
have so f
if not imp
cial imag
accuracy

In ou
[21] is e
it in the

2.4 Summary

Face recognition is a non-intrusive method for personal identification. It is widely accepted by people as a safe biometric technique and has the potential to become the most popular biometric technique. However, for face recognition to be useful in practice, a number of constraints need to be imposed. As of today, face recognition works best with static, controlled and well illuminated frontal facial images taken against a plain background. Under such conditions, the face images can be easily segmented from the background and the resulting intra-class variations are small. But if such constraints are imposed then the system looses its user-friendliness. Furthermore, there is no conclusive evidence that facial images are a reliable proof of identity and that systems based on face recognition can achieve an acceptable identification accuracy.

Ideally, a face recognition system should be able to automatically detect the presence of a face in an image, segment the face, if one is present, and be able to identify the face from a general view point. Research efforts to solve these difficult problems have so far met with very limited success. We thus conclude that it is very difficult if not impossible, to design an automatic personal identification system based on facial images alone, that is capable of achieving an acceptable degree of recognition accuracy in a general and unconstrained environment.

In our system the *eigenface* technique for face recognition by Turk and Pentland [21] is employed. We have further extended the recognition system so as to operate it in the verification mode.

Chapter 3

Fingerprint Verification

A fingerprint is a pattern of ridges and furrows that are formed on the tips of the finger due to the accumulation of dead and cornified cells. The individuality of a fingerprint is completely defined by the local characteristics of the ridge pattern (*minute* details) and their relationships. The process of fingerprint verification involves (*i*) fingerprint image acquisition, (*ii*) the extraction and representation of the local ridge characteristics and (*iii*) the matching of two ridge characteristic patterns so as to measure the similarity/dissimilarity between them. A block diagram of a fingerprint verification system is shown in Figure 3.1. We shall now discuss the details of the various steps involved.

3.1 Fingerprint Image Acquisition

Fingerprint images can be acquired either in an off-line fashion, in which case they are referred to as *inked fingerprints* or *latent fingerprints*, or they can be acquired

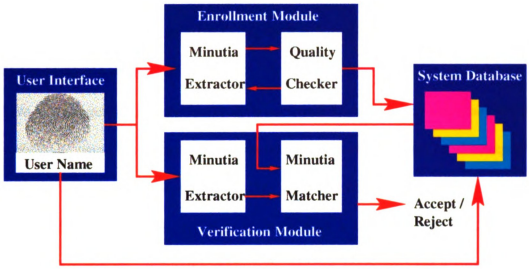


Figure 3.1: Architecture of the fingerprint verification system.

in an online fashion, in which case they are referred to as *live-scan fingerprints.* Although the inked method of fingerprint image acquisition is rather cumbersome and slow, it is has been the standard technique for over a hundred years. Furthermore, unavailability of direct feedback during the acquisition process makes it difficult to control the quality of the acquired images. Finally, this method is unacceptable in an automated verification system.

Live-scan fingerprints eliminate the intermediate step of getting an impression on paper by obtaining a digital image of the fingerprint directly from the finger. Thus, live-scan fingerprints are ideally suited for an automated verification system. A number of sensing mechanisms can be used for the acquisition of live-scan fingerprint images including (*i*) optical frustrated total internal reflection (FTIR), (*ii*) ultrasonic total internal reflection, (*iii*) optical total internal reflection of edge-lit holograms, (*iv*) thermal sensing of the temperature differential (across the ridges and valleys),

(r) sensit

of live-sc

currently

Live-s

cally obta

surface of

sists only

the time

valid ridg

deformati

and live-s

Figure 3

Of t

most po

concept

on the f

(*v*) sensing of differential capacitance, and (*vi*) non-contact 3D scanning. A number of live-scan fingerprint image acquisition devices based on the above techniques are currently available in the market.

Live-scan fingerprints vary markedly from inked fingerprints in that they are typically obtained using the *dabs* method, wherein a finger is impressed on the acquisition surface of a scanning device without rolling. Thus, a dabs live-scan fingerprint consists only of the ridge structure that is in contact with the acquisition surface at the time of acquisition. Although, such a fingerprint tends to have a smaller area of valid ridge structure as opposed to an inked fingerprint, it has a smaller amount of deformation than an inked fingerprint. Figure 3.2 shows examples of inked, latent and live-scan fingerprints.

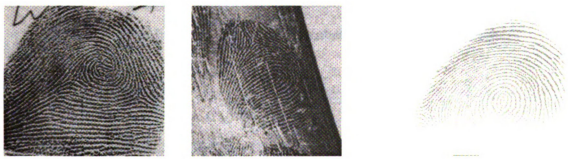


Figure 3.2: Different kinds of fingerprints: (a) inked, (b) latent, and (c) live-scan.

Of the various methods of obtaining live-scan fingerprints described above, the most popular techniques is based on optical frustrated total internal reflection (FTIR) concept. When a finger is placed on one side of a glass platen (prism), the ridges on the fingertip are in contact with the glass surface, whereas the furrows are not in

contact with the glass surface. A laser light source illuminates the glass at a certain angle. The points on the glass surface which are in contact with the ridges cause the incident light to be scattered randomly, whereas at the other points on the glass surface, the incident light suffers total internal reflection. The light reflected from the glass is captured by a CCD camera, resulting in a corresponding fingerprint image on the imaging plane of the CCD. Figure 3.3 shows the FTIR fingerprint scanner used in our verification system.

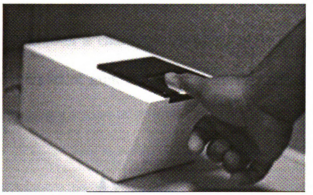


Figure 3.3: The live-scan fingerprint scanner from Digital Biometrics.

3.2 Feature Extraction

The role of feature extraction is to derive a set of representative features from the input image, that satisfy the following requirements: (*i*) *retain the discriminating power of the fingerprint image*, (*ii*) *compactness*, (*iii*) *suitable for use by a matching algorithm*, (*iv*) *robust to noise and distortions*, and (*v*) *easy to compute*. Since the individuality of fingerprints is completely determined by the local ridge characteristics, the first property postulates that a representation that captures these local characteristics is

best suited for use in automatic fingerprint verification systems. To date, a total of one hundred and fifty different local ridge characteristics, called *minute details*, have been identified. However, they are not evenly distributed and their detection depends on the image quality and impression conditions. The two most prominent ridge characteristics, called *minutiae*, are (*i*) *ridge ending* and (*ii*) *ridge bifurcation*. A ridge ending is defined as a point on the ridge at which a ridge terminates abruptly and a ridge bifurcation is defined as a point on the ridge where a ridge forks or diverges into branch ridges. Given a fingerprint image of reasonable quality, minutiae can be identified easily. Each minutia is characterized by its type, its x and y coordinates, and its direction. Examples of minutiae along with the features that characterize them are shown in Figure 3.4.

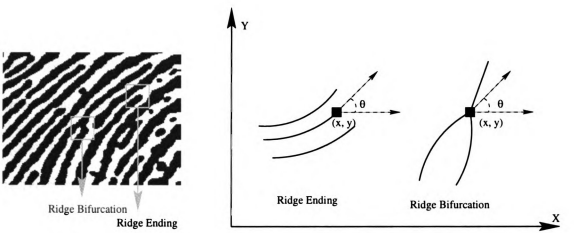


Figure 3.4: Examples of minutiae and their characterization.

A representation based on minutiae is compact, suitable for use by a matching algorithm, robust to noise and distortions, and easy to compute. Due to varying lighting conditions and due to variations in the impression pressure, a ridge ending may be

mistaken for a ridge bifurcation and vice versa, hence no distinction is made between the two kinds of minutiae. Each detected minutiae is, thus, characterized by the following three parameters: (*i*) *x-coordinate*, (*ii*) *y-coordinate*, and (*iii*) *orientation*.

3.2.1 Minutiae Extraction

For our fingerprint verification system, we make use of the minutiae extraction algorithm proposed in [25]. The overall flowchart of the algorithm is depicted in Figure 3.5. It mainly consists of three stages: (*i*) orientation field estimation, (*ii*) ridge extraction, and (*iii*) minutiae extraction and post-processing. First, local ridge orientation is estimated and the region of interest is located. Ridges are extracted from the input gray-level images, processed to get rid of the small speckles and holes, and thinned to obtain 8-connected single-pixel wide ridges. Finally, in the minutiae extraction stage, minutiae are extracted from the thinned ridges and refined using certain heuristics.

For each minutiae detected by the above algorithm, the following parameters are recorded: (*i*) x-coordinate, (*ii*) y-coordinate, (*iii*) orientation, which is defined as the local ridge orientation of the associated ridge, and (*iv*) the associated ridge segment. The recorded ridges are represented as one-dimensional discrete signals, normalized by a preset length parameter which is approximately equal to the average inter-ridge distance in the fingerprints. About 10 locations on the ridge associate with each minutiae are sampled. A fingerprint image when represented by the above representation and stored in a compressed format takes, on an average, about 250

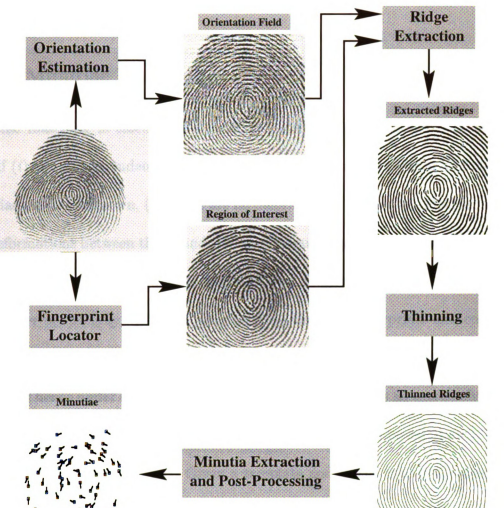


Figure 3.5: Flowchart of the minutiae extraction algorithm after Jain *et al.* [2].

bytes, a reduction by a factor of approximately 1,228 from the original size of 307,208 bytes for a 640 × 480 8-bit image.

3.3 Fingerprint Matching

The fingerprint matching problem is defined as follows: Given two (a test and a template) fingerprint representations, determine whether the two fingerprints are im-

pressions of the same finger [2].

3.3.1 Minutiae Matching

Minutiae matching is essentially a point pattern matching problem. In the ideal case, if (*i*) the correspondence between the template minutiae pattern and the input minutiae pattern is known, (*ii*) there are no deformations such as translation, rotation and deformations between them, and (*iii*) each minutiae present in a fingerprint image is exactly localized, then minutiae matching is a relatively simple task of counting the number of spatially matching pairs between the two fingerprints and comparing it against a pre-specified threshold value. These conditions are rarely satisfied in reality and hence under real life conditions, minutiae matching is an extremely difficult problem (refer to Figure 3.6). The difficulty of the problem can be attributed, mainly, to the following two reasons: (*i*) Given a template and a test minutiae pattern from the same finger, it is still necessary to establish the correspondence between the two, and (*ii*) the imaging system and the process of fingerprint image acquisition introduce a number of errors in the representation. In order for the minutiae matching algorithm to operate reliably in practice, it is necessary to establish and characterize a realistic model of the variations that occur among the representations of mated pairs. The following properties are desirable to be included in the model [25]:

1. The finger may be placed at different locations on the sensor resulting in a (global) translation between the minutiae from the test and the template representations.

Figure 3.6: Fingerprint matching problem: (a) and (b) are two different impressions from the same finger.

2. The finger may be placed in different orientations on the sensor resulting in a (global) rotation between the minutiae from the test and the template representations.

3. The finger may exert a different (average) downward normal pressure on the sensor resulting in a (global) spatial scaling between the minutiae from the test and the template representations.

4. The finger may exert a different (average) shear force on the sensor resulting in a (global) shear transformation (characterized by a shear direction and magnitude) between the minutiae from the test and the template representations.

5. Spurious minutiae may be present in both the template as well as the test representations.

6. Genuine minutiae may be absent in the template or test representations.

7. Minutiae may be locally perturbed from their true location and the perturbation

may be different for each individual minutiae.

8. The individual perturbations among the corresponding minutiae could be relatively large (with respect to ridge spacing) but the perturbations among pairs of minutiae are spatially linear.

9. The individual perturbations among the corresponding minutiae could be relatively large (with respect to ridge spacing) but the perturbations among pairs of minutiae are spatially non-linear.

10. Only a (ridge) connectivity preserving transformation could characterize the relationship between the test and template representations.

A large number of minutiae matching algorithms which are essentially "Euclidean" matchers have been proposed in the literature [26, 27, 28, 29, 30, 31]. These algorithms satisfy the above assumptions to varying degrees. However, they are either too slow for use in real-time systems or are not reliable enough in terms of accuracy. In our system an alignment-based matching algorithm developed in [25] is used. The algorithm is simple in theory, efficient in discrimination, and fast in speed. Given a template representation consisting of M minutiae denoted as $P = ((x_1^P, y_1^P, \theta_1^P)^T, \cdots, (x_M^P, y_M^P, \theta_M^P)^T)$ and an input representation consisting of N minutiae denoted as $Q = ((x_1^Q, y_1^Q, \theta_1^Q)^T, \cdots, (x_N^Q, y_N^Q, \theta_N^Q)^T)$, the algorithm performs minutiae matching by executing the following steps:

1. *Estimate the translation and rotation parameters between the ridge associated with each input minutiae and the ridge associated with each template minutiae*

and align the two minutiae patterns according to the estimated parameters.

2. *Convert the template pattern and input pattern into the polar coordinate representations with respect to the corresponding minutiae on which the alignment is achieved and represent them as two symbolic strings by concatenating each minutiae in an increasing order of radial angles:*

$$P_p = ((r_1^P, e_1^P, \theta_1^P)^T, \cdots, (r_M^P, e_M^P, \theta_M^P)^T) \quad (3.1)$$

$$Q_p = ((r_1^Q, e_1^Q, \theta_1^Q)^T, \cdots, (r_N^Q, e_N^Q, \theta_N^Q)^T), \quad (3.2)$$

where r_*, e_*, and θ_* represent the corresponding radius, radial angle, and normalized minutiae orientation with respect to the reference minutiae, respectively.

3. *Match the resulting strings P_p and Q_p with a modified dynamic programming algorithm to find the 'edit distance' between P_p and Q_p.*

4. *Use the minimum edit distance between P_p and Q_p to establish the correspondence of the minutiae between P_p and Q_p. The matching score, S, is then defined as:*

$$S = \frac{100 M_{PQ} M_{PQ}}{MN}, \quad (3.3)$$

where M_{PQ} is the number of minutiae which fall in the bounding boxes of tem-

plate minutiae. The bounding box of a minutiae specifies the possible positions of the corresponding input minutiae with respect to the template minutiae.

Figure 3.7 depicts the above algorithm.

3.4 Experimental Results in Literature

The performance of the algorithm discussed above has been evaluated using the MSU fingerprint database and a portion of the NIST 9 fingerprint database. The MSU database consists of 150 individuals with 10 fingerprints per individual. The fingerprints have been obtained using a live-scan fingerprint reader. No restrictions on the position and orientation of the finger were imposed. Approximately 90% of the fingerprints are estimated to be of fairly good quality. Jain *et al.* [2] report an authentic accept rate of 87.5% with a false accept rate of 0.01% for the MSU database. The NIST database consists of 1,350 mated fingerprint card pairs. Each mated pair consists of a fingerprint obtained using the rolled method and a fingerprint obtained using the live-scan method. The reported performance figures are 83.1% authentic accept rate with a false accept rate of 0.012% [2].

3.5 Summary

Fingerprints have been used for centuries for establishing the identity of an individual. Their biological properties are well understood and extensive research has been conducted on fingerprint matching. The uniqueness of a fingerprint is completely defined

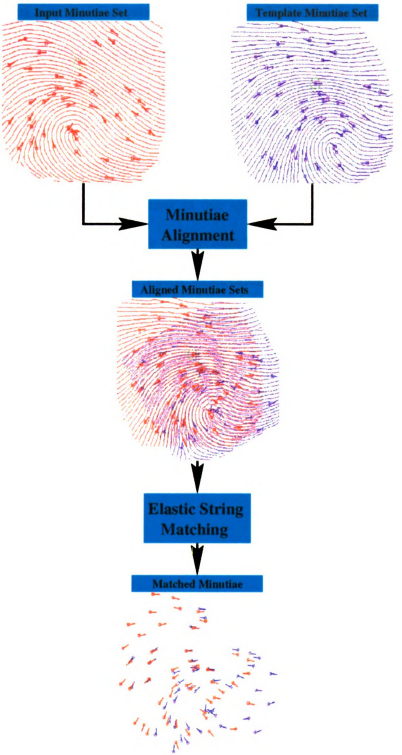


Figure 3.7: Flowchart of the minutiae matching algorithm.

by the local ridge characteristics. Minutiae, which are defined as ridge endings and ridge bifurcations, are the most prominent amongst the local ridge characteristics and a representation based on minutiae has been successful in designing automatic fingerprint verification systems. We have decided to use the minutiae extraction algorithm proposed in [25] because it is fast, efficient and tolerant to noise. Minutiae matching is essentially a point-pattern matching problem. The alignment-based elastic algorithm proposed in [25] has been used in our system for its speed and robustness. The algorithm is able to adaptively compensate for the nonlinear deformations and inexact transformations between mated fingerprints and hence is able to achieve a good verification accuracy at an acceptable speed.

Chapter 4

Speaker Verification

The speaker verification problem can be stated as follows: "Given an utterance, verify the identity claimed by the speaker against a database of known speakers." Depending upon the application, the speaker may be prompted (visually or orally) to speak a phrase known to the system. Alternatively, the system may attempt to verify the identity claim without any knowledge of the actual words that were spoken. The former mode of operation is termed as *text-dependent* speaker verification, whereas the latter is termed as *text-independent* speaker verification.

Text-dependent speaker verification systems can either make use of a small vocabulary and require the user to speak certain words selected at random from the vocabulary, or allow the user to select his "password" phrase and require him to use the same phrase each time. The use of a small vocabulary is generally preferred since the random choice of words to be spoken at the time of verification makes the system less circumventable by fraudulent means. Furthermore, it is also possible to build a speech recognition system for the chosen vocabulary which is then used to validate the

correctness of the acquired speech data at enrollment as well as during verification.

We have designed and implemented a text-dependent speaker verification system that uses a vocabulary of four digits: *one*, *two*, *seven*, and *nine*. This vocabulary was chosen for the following reasons: (*i*) Rabiner *et al.* [32] have demonstrated that *hidden Markov models* (HMMs) can be used for generating robust spoken digit models for an individual, and (*ii*) given four digits, 24 combinations are possible, thus, effectively increasing the length of the vocabulary from 4 to 24 which in turn makes the system less susceptible to tape recording fraud (the use of prerecorded speech). There are no particular reasons for the choice of the particular four digits, or for electing to have four digits instead of three or five. It can, however, be argued that the choice of the number of digits represents a trade-off between user convenience and susceptibility of the system to tape recording fraud: a large number of digits used in combinations would make the system practically impossible to break by tape recording fraud but would also increase the time required for enrollment as well as verification and hence could render the system unusable in a practical scenario.

Input to the system consists of a combination of the four digits, visually presented to the speaker on a video monitor. During enrollment, twelve of the twenty-four possible combinations of the four digits are recorded for each user. Each composite utterance is then segmented to obtain four sets of training data, one for each digit. Each set is then used to generate a spoken-digit model for that speaker. Thus, four models are generated for each speaker. The training samples from all the users are then pooled to train a speech recognition system. During verification, the user is prompted visually to speak a combination chosen at random by the system from the

24 possible combinations. The composite utterance is segmented into four utterances and each utterance is fed to the speech recognition system. If the four utterances are correctly recognized as the prompted combination, they are compared with the corresponding spoken-digit models for the user and a matching score is computed. The matching score is then fed to the decision fusion module which then makes the final accept/reject decision in conjunction with the results of the face verification system and the fingerprint verification system. Figure 4.1 depicts a block diagram of our speaker verification system. It consists of three modules: (*i*) speech acquisition and feature extraction, (*ii*) speech model generation, and (*iii*) pattern matching.

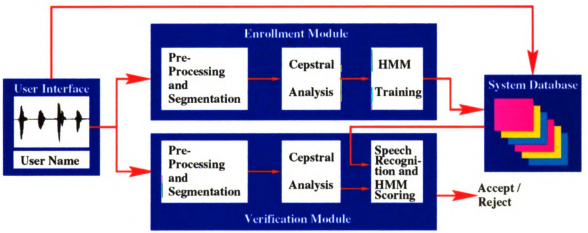


Figure 4.1: A block diagram of the HMM-based Speaker Verification Module.

4.1 Feature Extraction

The speaker-specific characteristics of speech are due to differences in physiological and behavioral aspects of the speech-production system in humans [33]. The main physiological aspect of the human speech production system is the vocal tract

shape. The vocal tract is generally considered as the speech production organs above the vocal folds, which consists of the following: (*i*) laryngeal pharynx (beneath the epiglottis), (*ii*) oral pharynx (behind the tongue, between the epiglottis and velum), (*iii*) oral cavity (forward of the velum and bounded by the lips, tongue, and palate), (*iv*) nasal pharynx (above the velum, rear end of nasal cavity), and (*v*) nasal cavity (above the palate and extending from the pharynx to the nostrils). The shaded area in Figure 4.2 depicts the vocal tract [3].

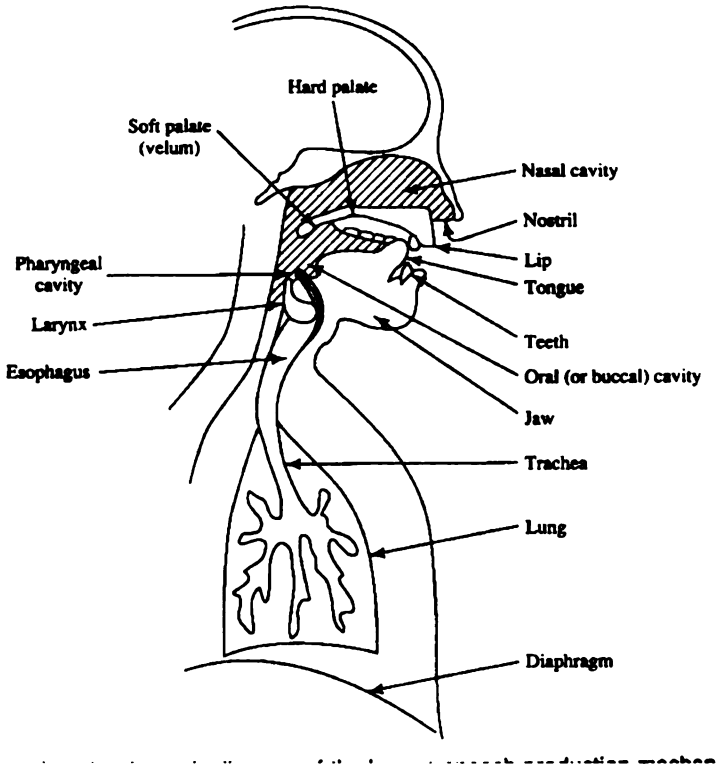


Figure 4.2: A schematic diagram of the human speech production mechanism [3].

The salient acoustic features of speech are contained in the spectral modifications made to the source excitation by the vocal tract system. Hence, it is common in speaker verification systems to make use of features derived only from the vocal tract.

In order to characterize the features of the vocal tract, the human speech production mechanism is represented as a discrete-time system of the form depicted in Figure 4.3 [3].

The acoustic wave is produced when the airflow from the lungs is carried by the trachea through the vocal folds. This source of excitation can be characterized as phonation, whispering, frication, compression, vibration, or a combination of these [3]. Phonated excitation occurs when the airflow is modulated by the vocal folds. Whispered excitation is produced by airflow rushing through a small triangular opening between the arytenoid cartilage at the rear of the nearly closed vocal folds. Frication excitation is produced by constrictions in the vocal tract. Compression excitation results from releasing a completely closed and pressurized vocal tract. Vibration excitation is caused by air being forced through a closure other than the vocal folds, especially at the tongue. Speech produced by phonated excitation is called *voiced*, that produced by phonated excitation plus frication is called *mixed*, and that produced by other types of excitation is called *unvoiced*.

From Figure 4.3, the overall transfer function of the speech production system can now be given as

$$\Theta(z) = \frac{S(z)}{E(z)} = \begin{cases} H(z)R(z) & \text{unvoiced case} \\ G(z)H(z)R(z) & \text{voiced case.} \end{cases} \tag{4.1}$$

Thus, it is possible to represent the vocal-tract in a parametric form as the trans-

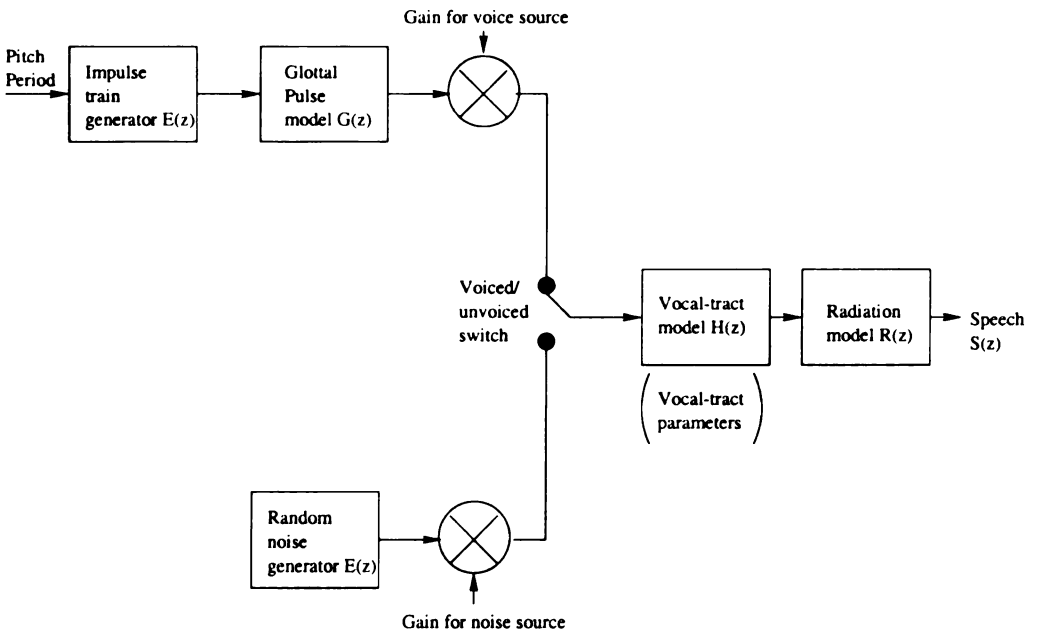


Figure 4.3: A discrete-time model representing the human speech production system [3].

fer function $H(z)$. In order to estimate the parameters of $H(z)$ from the observed speech waveform, it is necessary to assume some form for $H(z)$. Ideally, the transfer function should contain poles as well as zeros. However, if only the voiced regions of speech are used then an all-pole model for $H(z)$ is sufficient. Furthermore, linear prediction analysis can be used to efficiently estimate the parameters of an all-pole model. Finally, it can also be noted that the all-pole model is the minimum-phase part of the true model and has a magnitude spectrum that is identical to the magnitude spectrum of the true model, which contains the bulk of the speaker dependent information. Figure 4.4 illustrates the differences in the models for different sounds spoken by the same individual [3]. Figure 4.5 illustrates the differences in the models for two speakers saying the same vowel.

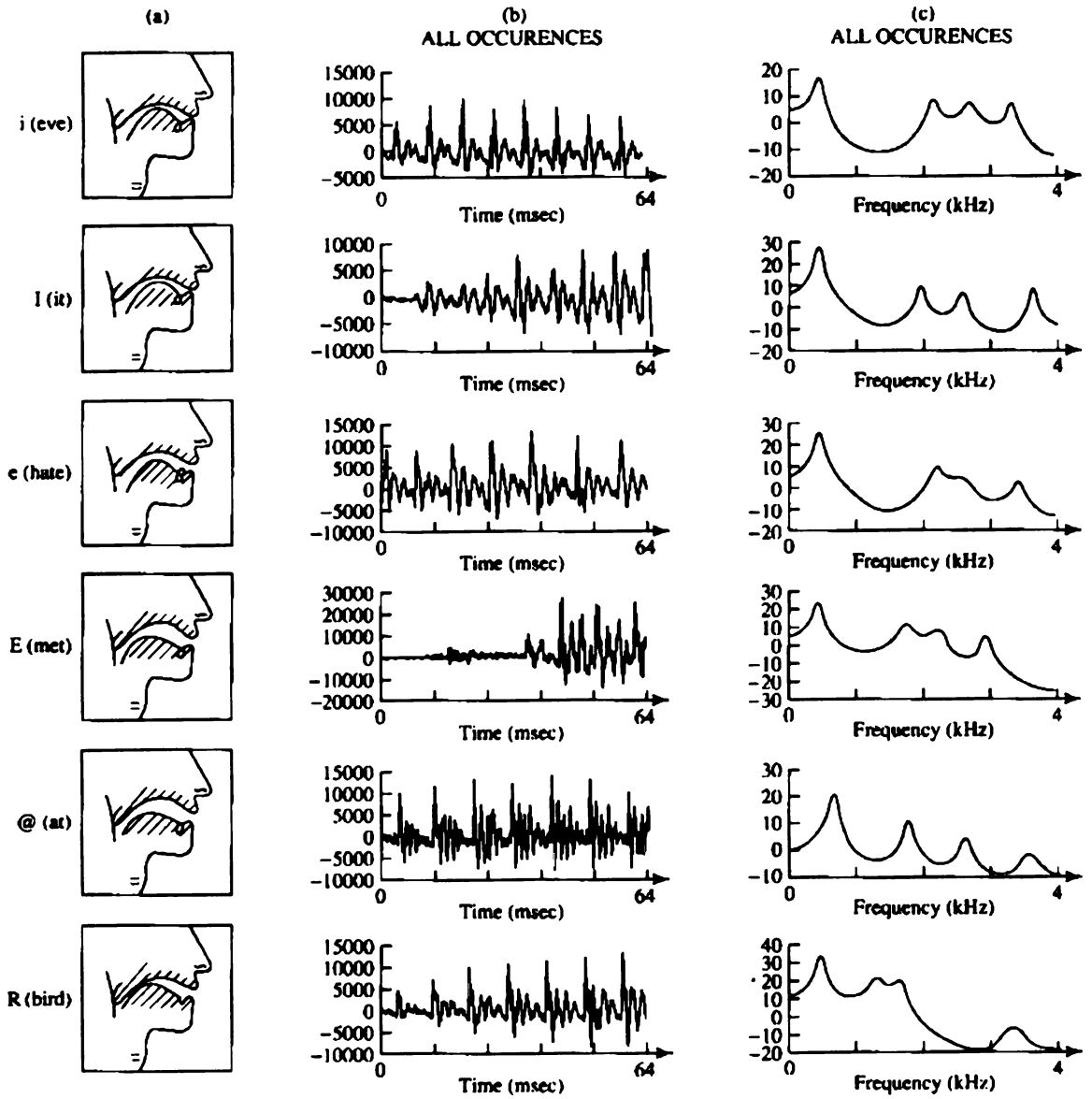


Figure 4.4: Vocal-tract models for different sounds [3].

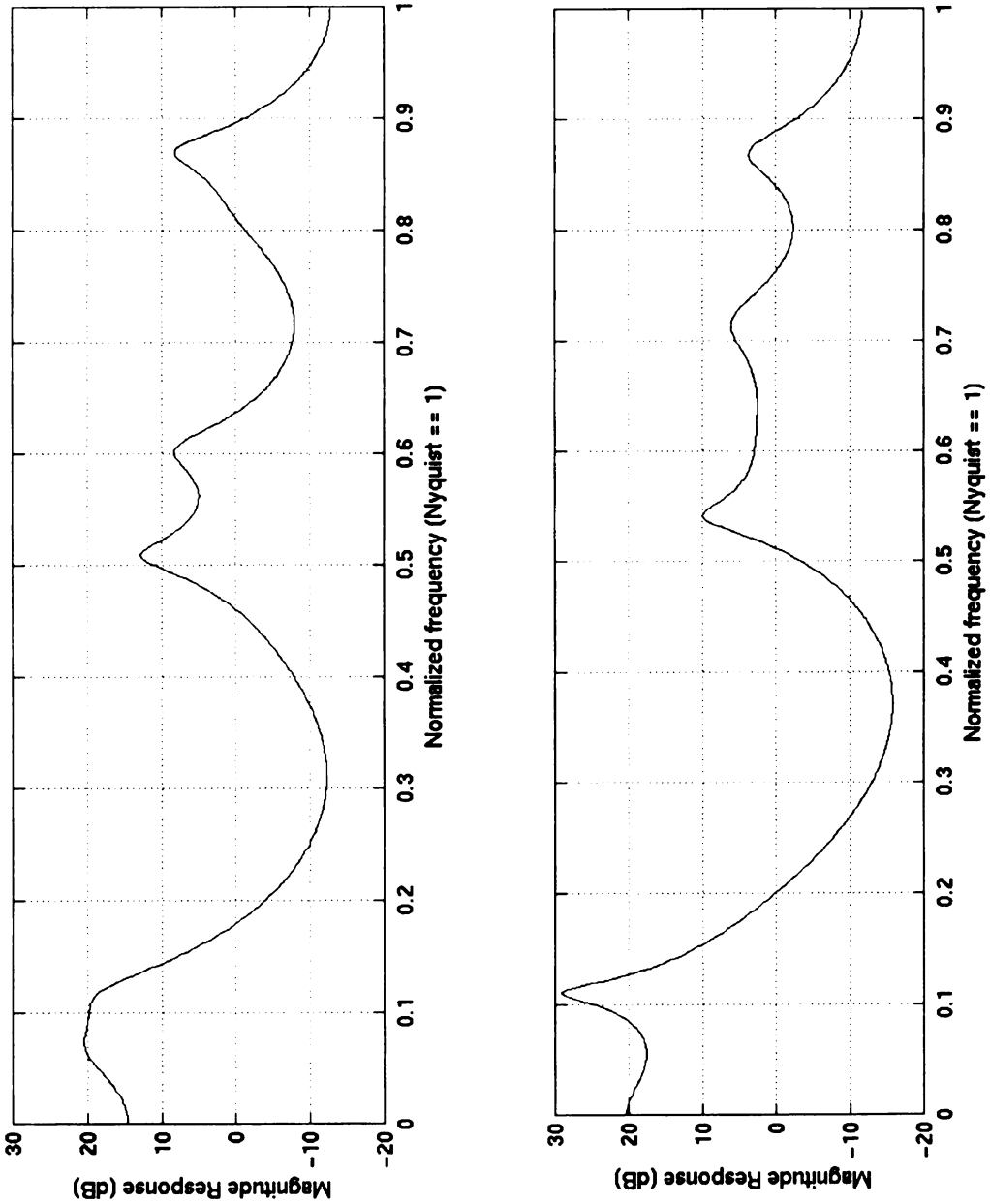


Figure 4.5: Vocal-tract magnitude spectrum for letter 'a' for two different speakers.

4.1.1 Speech Acquisition and Preprocessing

The spoken speech is converted to an analog signal through the use of a noise cancellation headset microphone manufactured by Labtec. The analog speech signal is digitized using a Creative Labs SoundBlaster audio card. The sampling rate used is 8KHz and the resolution is 16 bits/sample. The digitized speech signal, $s(n)$, is then processed by a first-order FIR filter to spectrally flatten the signal and to make it less susceptible to finite precision effects during the later stages of processing. The z-transform of the first-order FIR filter used in our system is

$$H(z) = 1 - az^{-1}, \tag{4.2}$$

where $a = 0.95$. Thus, the preemphasized signal $\tilde{s}(n)$ is related to $s(n)$ as,

$$\tilde{s}(n) = s(n) - as(n-1). \tag{4.3}$$

The preemphasized signal, $\tilde{s}(n)$, is then blocked into frames of $N = 300$ samples, with a shift of $M = 100$ samples. This corresponds in time to 37.5 ms frames with a 12.5 ms shift between frames. If the lth frame of speech is denoted by $x_l(n)$ and there are L frames within the entire speech signal then

$$x_l(n) = \tilde{s}(Ml + n), \qquad n = 0, 1, \cdots, N-1, \qquad l = 0, 1, \cdots, L-1. \tag{4.4}$$

Each frame is then weighted by a Hamming window,

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N}\right), \qquad 0 \leq n \leq N-1. \tag{4.5}$$

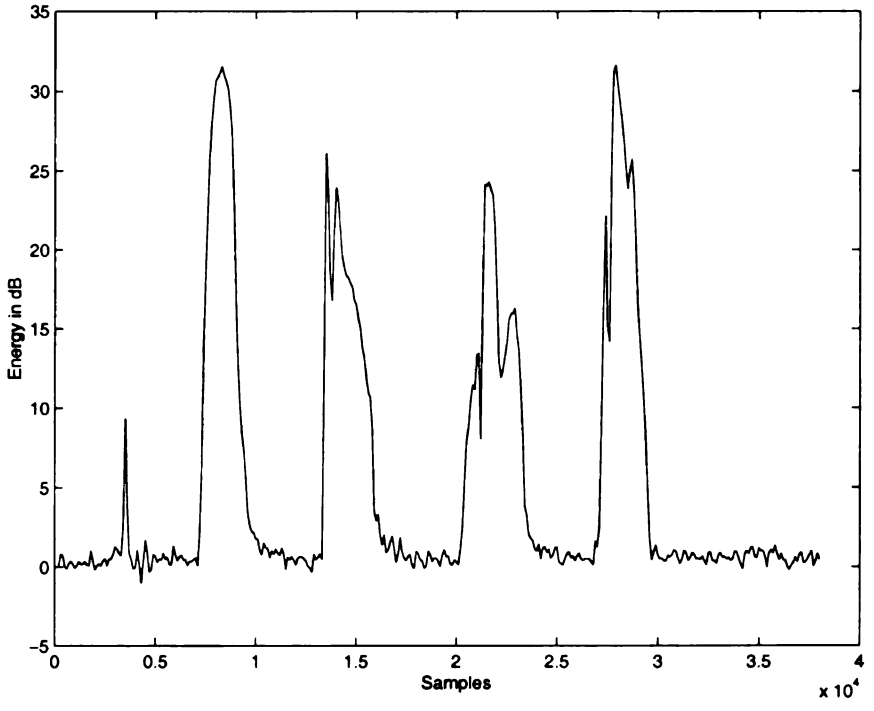
The windowed signal is

$$\tilde{x}_l(n) = x_l(n) w(n), \qquad 0 \leq n \leq N-1. \tag{4.6}$$

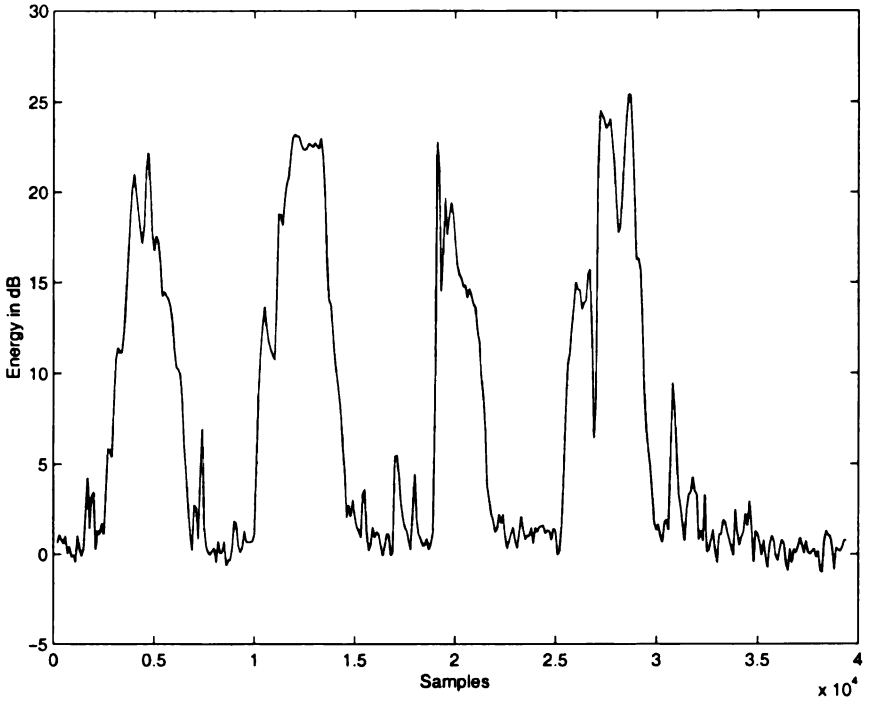
4.1.2 Speech Segmentation

Since the speech signal is a composite utterance consisting of four digits, it is segmented so as to obtain the four sub-utterances corresponding to the four digits. In order to segment the speech, we compute the log energy in dB (Equation 4.7) of each frame. A typical plot of the log energy for an utterance consisting of four digits is shown in Figure 4.6(a). We note that corresponding to each spoken digit, there is a peak in the energy waveform. Thus, in the absence of background noise and other speech artifacts, segmentation into digits is a relatively easy task [34].

However, in most practical scenarios, a fair amount of background noise and speech



(a) Background noise minimal



(b) Background noise significant

Figure 4.6: Plots of the log energy for speech acquired under different environments.

artifacts such as clicking while opening the mouth to speak, breath noise, etc. are present. These noise sources result in an energy waveform as shown in Figure 4.6(b). Under such conditions, the segmentation problem is no longer trivial [35].

We employ a modified version of the top-down approach proposed by Wilpon *et al.* [35] to detect the word boundaries. A brief description of the algorithm is given below, and the flowchart is shown in Figure 4.7.

The log energy of the speech signal is computed for each frame as

$$E(l) = 10 \log_{10} \left(\sum_{n=0}^{N-1} [\tilde{x}_l(n)]^2 \right), \qquad l = 0, 1, \cdots, L-1. \tag{4.7}$$

In order to compensate for the background noise, *adaptive-level equalization* of the energy contour is performed. First, E_{min} is computed as

$$E_{min} = \min_{0 \le l < L} E(l)$$

which is then subtracted from the energy contour to obtain $\hat{E}(l)$.

$$\hat{E}(l) = E(l) - E_{min}, \qquad l = 0, 1, \cdots, L-1.$$

Next, a histogram of the signal energies below 10 dB is computed. This histogram is smoothed by applying a three-point median filter and the mode of the smoothed

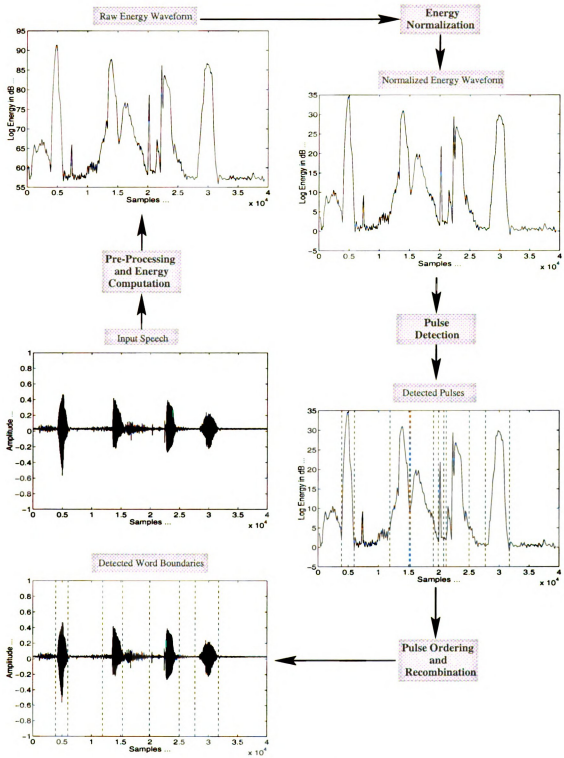


Figure 4.7: The flowchart of the algorithm for word boundary detection.

histogram is computed. Finally, the modified energy contour is computed as

$$\tilde{E}(l) = \hat{E}(l) - Mode.$$

Under ideal recording conditions, the resulting energy contour has the property that during periods of silence, the energy level oscillates around the 0 dB level (Figure 4.6(a)). However, in the presence of significant background noise, this may not be true (Figure 4.6(b)).

In the original algorithm [35] the equalized energy contour is scanned repeatedly for the maximum energy pulse. Next, the algorithm examines the frames to the left and the right of the frame with the peak energy until it finds frames in which the energy falls below a predefined threshold. Finally, the algorithm refines the pulse boundaries by checking the first few and the last few frames of the pulse for consistently low energy content. The choice of the predefined threshold is the key to the success of the algorithm. For our particular setup, a threshold below 5 dB needs to be chosen for the algorithm to work properly. However, we have observed that in the presence of widely variable background noise, any threshold below the 5 dB level gives inconsistent results. As a result, we have modified the algorithm as follows:

Once the peak energy frame is detected, we scan the energy frames to the left and the right of the frame. However, we chose a high threshold of 10 dB to obtain the initial estimates of the pulse boundaries. Next we examine a certain number of frames beyond the left and the right boundaries and record the minimum energies

in those intervals. The final pulse boundaries are then taken as the frames where the minimum energies occur. Furthermore, the choice of the number of frames to be examined is decided by the pulse width at the 10 dB level. Thus, for small steep pulses we examine a fewer number of frames than for large and wide pulses. This eliminates the need for selecting a threshold that may be sensitive to the background noise. This modified algorithm has been observed to give better results than the original algorithm.

The detected pulses are then checked for a minimum width and magnitude and are eliminated from the signal. The algorithm stops scanning the energy contour if the last peak detected was below a threshold (15 dB in our case).

The validated pulses are then arranged in a decreasing order of their peak amplitudes. Since each utterance is supposed to contain four digits, if the number of detected pulses is greater than four then an attempt is made to combine some of the pulses to form longer pulses. Starting with the highest peak amplitude pulse, the end points of the adjacent pulses are examined. If an adjacent pulse is found within a certain number of frames from the end point of the pulse under consideration, the two pulses are combined. If the number of pulses beyond this point is still greater than four, then the magnitudes of the pulses beyond the fourth largest pulse are checked and a decision is made either to discard the pulses or to reject the recording.

The log energy waveform of a typical utterance of the four digits *one*, *two*, *seven*, and *nine* along with the detected word boundaries is shown in Figure 4.8.

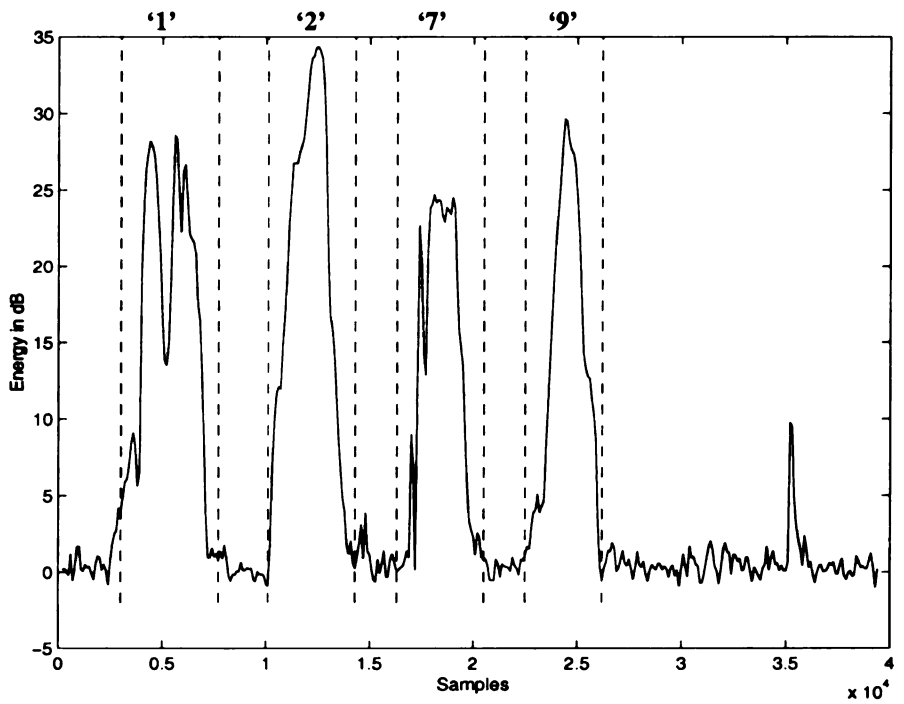


Figure 4.8: Output of the speech segmentation algorithm.

4.1.3 Cepstral Analysis

As discussed in Section 4.1, during a stationary frame of speech, the vocal-tract is generally modeled as an all-pole filter whose transfer function, $H(z)$, is given as

$$H(z) = \frac{H_0}{1 - \sum_{i=1}^{N} a(i) z^{-i}}, \tag{4.8}$$

where H_0 represents an overall gain term and $a(i)$ are the filter coefficients. The filter coefficients can be estimated to the pth order by using linear prediction analysis[3]. The coefficients thus estimated are called a *linear predictive code*(LPC).

The LPC features were very popular in the early speech-recognition and speaker-

verification systems. However, comparison of two LPC feature vectors requires the use of computationally expensive similarity measures such as the *Itakura-Saito* distance [32] and hence LPC features are unsuitable for use in real-time systems. Furui [36] suggested the use of *the cepstrum*, defined as the inverse Fourier transform of the logarithm of the magnitude spectrum, in speech-recognition applications. The use of the cepstrum allows for the similarity between two cepstral feature vectors to be computed by using the Euclidean distance. Furthermore, Atal [37] has demonstrated that the cepstrum *derived* from the LPC features results in the best performance in terms of FAR and FRR for speaker verification. Consequently, we use *the LPC derived cepstrum* for our speaker verification system. The order, p, of the LPC analysis is a parameter whose choice depends upon the application. For spoken digit recognition, Rabiner *et al.* [38] have demonstrated that $p = 10$ results in the best performance. Hence, for each frame a 10th order LPC analysis using Durbin's recursive method is performed to obtain 10 LPC coefficients. Durbin's recursive method consists of the following steps:

1. The autocorrelation coefficients up to the 10th order of each frame of the windowed signal are obtained as,

$$r_l(m) = \sum_{n=0}^{N-1-m} \tilde{x}_l(n)\tilde{x}_l(n+m), \qquad m = 0, 1, \cdots, 10. \tag{4.9}$$

2. Each frame of 11 autocorrelations is converted into 10 LPC coefficients by the

following recursive algorithm:

$$E^{(0)} = r(0), \tag{4.10}$$

$$k_i = \left\{ r(i) - \sum_{j=1}^{L-1} \alpha_j^{i-1} r(|i-j|) \right\} / E^{(i-1)}, \qquad i = 1, \cdots, 10 \tag{4.11}$$

$$\alpha_i^{(i)} = k_i, \tag{4.12}$$

$$\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)}, \tag{4.13}$$

$$E^{(i)} = (1 - k_i^2) E^{(i-1)}, \tag{4.14}$$

where the summation in Equation (4.11) is omitted for $i = 1$. The above set of equations is solved recursively for $i = 1, 2, \cdots, 10$ and the resulting 10 LPC coefficients are given as

$$a_m = \alpha_m^{(10)}, \qquad m = 1, \cdots, 10. \tag{4.15}$$

Next, Q cepstral coefficients, $c(m), m = 1, 2, \cdots, Q$, are computed from the $p = 10$ LPC coefficients, using the following recursion:

$$c_0 = ln\sigma^2 \tag{4.16}$$

$$c_m = a_m + \sum_{k=1}^{m-1} \left(\frac{k}{m}\right) c_k a_{m-k}, \qquad m = 1, \cdots, p \tag{4.17}$$

$$c_m = \sum_{k=1}^{m-1} \left(\frac{k}{m}\right) c_k a_{m-k}, \qquad m > p, \tag{4.18}$$

where σ^2 is the gain term in the LPC model. Generally, Q is chosen to be greater than p and is typically given as $Q \simeq (\frac{3}{2})p$ [38]. We choose $Q = 12$. This vector is then weighted by a window $W(m)$ of the form

$$W(m) = 1 + \frac{Q}{2} \sin\left(\frac{\pi m}{Q}\right), \qquad m = 1, \cdots, Q.$$

Finally, the time derivative of the sequence of weighted cepstral vectors is approximated by a first-order orthogonal polynomial over a finite length window of $(2K+1)$ frames ($K = 2$ in our case) to obtain 12 delta cepstrum coefficients. The 12 weighted cepstral coefficients are combined with the 12 delta cepstrum coefficients to form the feature vector for the given frame. The entire speech signal is thus represented by a sequence of feature vectors.

4.2 Speaker Modeling

Speech produced by the same person at different times result in similar, yet different, sequences of feature vectors. The purpose of speech modeling is to build models that capture these variations in the extracted set of features. There are two types of models that have been used extensively in speaker verification and speech recognition systems: *stochastic* models and *template* models. The stochastic model treats the speech production process as a parametric random process and assumes that the parameters of the underlying stochastic process can be estimated in a precise, well-defined manner. The template model attempts to model the speech production process in a non-parametric manner by retaining a number of sequences of feature vectors derived from multiple utterances of the same word by the same person. Template models dominated early work in speaker verification and speech recognition because the template model is intuitively more reasonable. However, recent work with stochastic models has demonstrated that these models are more flexible and result in better models of the speech production process [33].

A very popular stochastic model for modeling the speech production process is the hidden Markov model (HMM). HMMs are extensions to the conventional Markov models, wherein the observations are a probabilistic function of the state, i.e., the model is a doubly embedded stochastic process where the underlying stochastic process is not directly observable (it is hidden). The HMM can only be viewed through another set of stochastic processes that produce the sequence of observations. Thus, HMM is a finite-state machine, where a probability density function $p(x|s_i)$ is associ-

ated with each state s_i. The states are connected by a transition network, where the state transition probabilities are $a_{ij} = P(s_i|s_j)$. A fully connected three-state HMM is depicted in Figure 4.9.

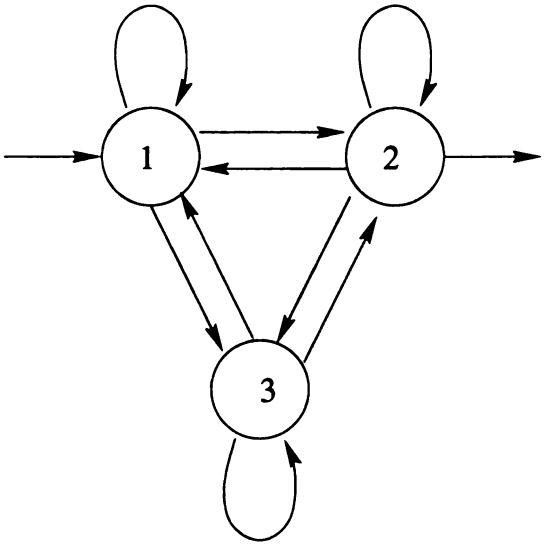


Figure 4.9: A fully connected three-state hidden Markov model.

For speech signals, another type of HMM, called a left-right model or a Bakis model, is found to be more useful. A left-right model has the property that as time increases, the state index increases (or stays the same)– that is the system states proceed from left to right. Since the properties of a speech signal change over time in a successive manner, this model is very well suited for modeling the speech production process. The parameters required for a complete specification of the HMM depicted in Figure 4.10 are summarized below. Again, the choice of these parameters depends upon the application and the particular choice of parameters, given below, has been demonstrated by Rabiner *et al.* [38] to result in the best performance for isolated spoken-digit recognition.

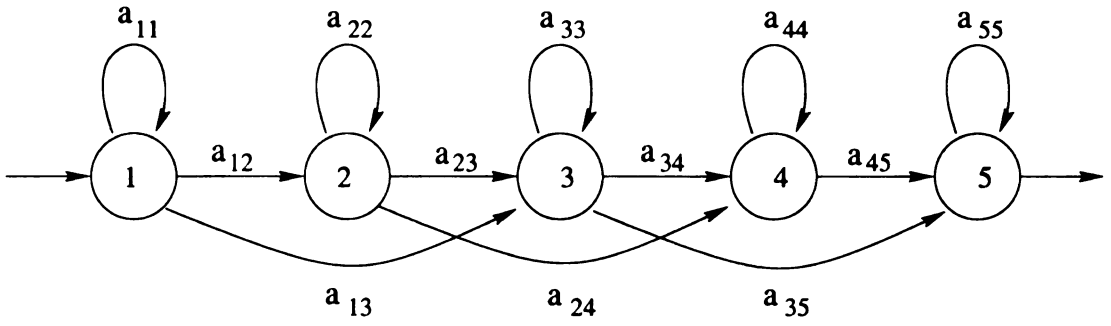


Figure 4.10: A left-to-right five state Hidden Markov Model.

1. N, the number of states in the model. In our case $N = 5$.

2. $A = [a_{ij}], i, j = 1, \cdots, N$, the state-transition matrix, where a_{ij} is the probability of making a transition from state j to state i. For a left-to-right model, we have the constraint $a_{ij} = 0, j < i, j > i + 2$ and $a_{NN} = 1$. The state-transition matrix is thus of the form

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 & 0 \\ 0 & a_{22} & a_{23} & a_{24} & 0 \\ 0 & 0 & a_{33} & a_{34} & a_{35} \\ 0 & 0 & 0 & a_{44} & a_{45} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.19}$$

3. $\pi = \{\pi_i\}, i = 1, \cdots, N$, the initial state distribution, in which $\pi_i = P[q_1 = i]$. Again, for a left-to-right model the initial state must have unity index and hence we have,

$$\pi_i = \begin{cases} 1 & \text{if } i = 1, \\ 0 & \text{if } i \neq 1. \end{cases} \tag{4.20}$$

4. B, the observation-probability function. For our case, the observation-probability function for each state is a continuous mixture density of the form

$$b_i(\mathbf{o}) = \sum_{k=1}^{M} c_{ik} \mathcal{N}(\boldsymbol{\mu}_{ik}, \mathbf{U}_{ik}), \qquad i = 1, \cdots, N, \tag{4.21}$$

where $\mathbf{o}$ is the 24-dimensional observation vector of cepstral coefficients, c_{ik} is the mixture coefficient for the kth mixture in state i and $\mathcal{N}$ is a 24-dimensional multivariate Gaussian density with mean vector $\boldsymbol{\mu}_{ik}$ and covariance matrix $\mathbf{U}_{ik}$ for the kth mixture component in state i. The mixture gains c_{ik} satisfy the following constraints:

$$\sum_{k=1}^{M} c_{ik} = 1, \qquad i = 1, \cdots, N \tag{4.22}$$

$$c_{ik} \geq 0, \qquad 1 \leq i \leq N, \qquad k = 1, \cdots, M \tag{4.23}$$

so that the probability density function is properly normalized, i.e.,

$$\int_{-\infty}^{\infty} b_j(\mathbf{o})d\mathbf{o} = 1, \qquad j = 1, \cdots, N. \tag{4.24}$$

We choose $M = 5$ mixtures per state.

The complete parameter set of the model, as described above, is denoted compactly as

$$\lambda = (A, B, \pi). \tag{4.25}$$

Given an HMM with a parameter set λ, we can now compute the probability that a particular sequence of observation vectors, denoted as

$$\mathbf{O} = (\mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_T) \tag{4.26}$$

was generated by the model. This conditional probability is denoted as $P(\mathbf{O}|\lambda)$. The training data for each digit is used to estimate the parameter set λ for the corresponding HMM, so as to maximize $P(\mathbf{O}|\lambda)$. Before we discuss the training procedure, we will first list some variables and describe procedures for evaluating these variables [39]:

1. *The forward variable*, $\alpha_t(i)$, is defined as

$$\alpha_t(i) = P(\mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_t, q_t = i|\lambda), \tag{4.27}$$

which is the probability of the partial observation sequence, $\mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_t$, (until time t) and state i at time t, given the model λ. $\alpha_t(i)$ is evaluated inductively as follows:

(a) Initialization

$$\alpha_1(i) = \pi_i b_i(\mathbf{o}_1), \qquad i = 1, \cdots, N. \tag{4.28}$$

(b) Induction

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(\mathbf{o}_{t+1}), \qquad \begin{array}{l} t = 1, \cdots, T-1 \\ j = 1, \cdots, N. \end{array} \tag{4.29}$$

The above procedure is referred to as the *forward* procedure [39].

2. *The backward variable*, $\beta_t(i)$, is defined as

$$\beta_t(i) = P(\mathbf{o}_{t+1}, \mathbf{o}_{t+2}, \cdots, \mathbf{o}_T|q_t = i, \lambda) \tag{4.30}$$

which is the probability of the partial observation sequence from $t+1$ to the end, given state i at time t and the model λ. $\beta_t(i)$ is evaluated inductively as follows:

(a) Initialization

$$\beta_T(i) = 1, \qquad i = 1, \cdots, N. \tag{4.31}$$

(b) Induction

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(\mathbf{o}_{t+1}) \beta_{t+1}(j), \qquad \begin{array}{l} t = T-1, T-2, \cdots, 1 \\ i = 1, \cdots, N. \end{array} \tag{4.32}$$

The above procedure is referred to as the *backward* procedure [39].

3. *The posteriori probability variable*, $\gamma_t(i,k)$, is defined as

$$\gamma_t(i,k) = P(q_t = i | \mathbf{O}, \lambda), \tag{4.33}$$

which is the probability of being in state i at time t with the kth mixture component accounting for $\mathbf{o}_t$, given the observation sequence $\mathbf{O}$, and the model λ. $\gamma_t(i,k)$ is evaluated using $\alpha_t(i)$ and $\beta_t(i)$ as follows [39]:

$$\gamma_t(i,k) = \left[\frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N}\alpha_t(i)\beta_t(i)}\right]\left[\frac{c_{ik}\mathcal{N}(\mu_{ik},\mathbf{U}_{ik})}{\sum_{m=1}^{M}c_{im}\mathcal{N}(\mu_{im},\mathbf{U}_{im})}\right]. \tag{4.34}$$

Rabiner *et al.* [39] also define $\gamma_t(j)$ as the probability of being in state j at time t given the observation sequence $\mathbf{O}$, and the model λ. From the definition of $\gamma_t(j,k)$

$$\gamma_t(j) = \sum_{k=1}^{M}\gamma_t(j,k). \tag{4.35}$$

Based on the above definitions, the training procedure for a HMM, also known as the *Baum-Welch* [40] method is given as follows:

First define $\xi_t(i,j)$ as the probability of being in state i at time t, and state j at time $t+1$, given the model and the observation sequence, i.e.,

$$\xi_t(i,j) = P(q_t = 1, q_{t+1} = j|\mathbf{O},\lambda). \tag{4.36}$$

$\xi_t(i,j)$ can be written in terms of the forward and backward variables as

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(\mathbf{o}_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}b_j(\mathbf{o}_{t+1})\beta_{t+1}(j)}. \tag{4.37}$$

Since $\gamma_t(i)$ is the probability of being in state i at time t, given the entire observation sequence and the model, we have

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j). \tag{4.38}$$

The summation over t (from $t = 1$ to $t = T-1$) of $\gamma_t(i)$ is the expected number of transitions from state i in the observation sequence $\mathbf{O}$. The summation over t (from $t = 1$ to $t = T-1$) of $\xi_t(i,j)$ is the expected number of transitions from state i to state j.

Using the above formulas and an initial estimate of the parameters, re-estimation of the parameters is performed by the following equations:

$$\begin{aligned}\bar{\pi}_i &= \text{number of times in state } i \text{ at time } (t=1) \\ &= \gamma_1(i) \end{aligned} \tag{4.39}$$

$$\begin{aligned}\bar{a}_{ij} &= \frac{\text{expected number of transitions from state } i \text{ to state } j}{\text{expected number of transitions from state } i} \\ &= \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \end{aligned} \tag{4.40}$$

$$\begin{aligned}\bar{c}_{ik} &= \frac{\text{expected number of times the system is in state } i \text{ using mixture } k}{\text{expected number of times the system is in state } i} \\ &= \frac{\sum_{t=1}^{T} \gamma_t(i,k)}{\sum_{t=1}^{T} \sum_{k=1}^{M} \gamma_t(i,k)} \end{aligned} \tag{4.41}$$

$$\bar{\boldsymbol{\mu}}_{ik} = \frac{\sum_{t=1}^{T} \gamma_t(i,k) \cdot \mathbf{o}_t}{\sum_{t=1}^{T} \gamma_t(i,k)} \tag{4.42}$$

$$\bar{\mathbf{U}}_{ik} = \frac{\sum_{t=1}^{T} \gamma_t(i,k) \cdot (\mathbf{o}_t - \boldsymbol{\mu}_{ik})(\mathbf{o}_t - \boldsymbol{\mu}_{ik})^t}{\sum_{t=1}^{T} \gamma_t(i,k)} \tag{4.43}$$

Thus, given an initial model estimate $\lambda = (A, B, \pi)$, Equations. (4.39)-(4.43) result in a re-estimated model denoted as $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$. It has been proven by Baum and his colleagues [40] that either (i) the initial model λ defines a critical point of the likelihood function, in which case $\bar{\lambda} = \lambda$; or (ii) model $\bar{\lambda}$ is more likely than model λ in the sense that $P(\mathbf{O}|\bar{\lambda}) > P(\mathbf{O}|\lambda)$. Thus the training procedure proceeds by iteratively using $\bar{\lambda}$ in place of λ and repeating the re-estimation calculations. The re-estimation procedure is terminated when $(P(\mathbf{O}|\bar{\lambda}) - P(\mathbf{O}|\lambda)) < \epsilon$, where ϵ is some

predefined threshold.

As with any iterative technique, a good choice of initial parameters is imperative in order to ensure the convergence of the re-estimation procedure. A number of algorithms for obtaining good initial estimates of the parameters have been proposed. For our system, we make use of the algorithm proposed in [32]. The main steps in the algorithm are summarized below:

1. *Choose the state transition matrix such that the sum of each row is 1.0.*

2. *For a given set of utterances, compute the global mean and variance of all the feature vectors and use this global mean and variance as the mean and variance of the mixture densities for each state.*

3. *Use the Viterbi algorithm [41] to segment the utterances into states.*

4. *For each state, apply a clustering algorithm on the feature vectors assigned to that state so as to obtain the same number of clusters as the number of components in the mixture.*

5. *Update the mean and variances of the mixture densities to be the mean and variances of the corresponding clusters.*

6. *Repeat steps 3 through 5 till the change in the model parameters between two successive updates is less than a predefined value (ϵ).*

The training procedure described above is also used in our system to train four speaker-independent models corresponding to the four digits. These speaker indepen-

dent models are used for performing speech recognition and also for normalizing the resulting matching score as described in the next section.

4.3 Pattern Matching

The pattern matching process involves the comparison of a given set of input feature vectors against the speaker model for the claimed identity and computing a matching score. For the hidden Markov models discussed above, the matching score is the probability that a given set of feature vectors was generated by the model. Let the set of feature vectors corresponding to the input speech be denoted as $\mathbf{O} = (\mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_T)$ and let the HMM be denoted as $\lambda = (A, B, \pi)$. We wish to compute the conditional probability $P(\mathbf{O}|\lambda)$. From the definition of the forward variable $\alpha_T(i)$, this conditional probability is simply given as

$$P(\mathbf{O}|\lambda) = \sum_{i=1}^{N} \alpha_T(i). \tag{4.44}$$

However, in most cases we are interested in determining the optimal sequence of states for the given observation sequence and the model and then computing the probability of that state sequence. The Viterbi algorithm [41] is used to find the single best state sequence, $\mathbf{q} = (q_1 q_2 \cdots q_T)$, for the given observation sequence $\mathbf{O} = (\mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_T)$ and the HMM $\lambda = (A, B, \pi)$ as follows

First the quantity $\delta_t(i)$ is defined as

$$\delta_t(i) = \max_{q_1,q_2,\cdots,q_{t-1}} P[q_1 q_2 \cdots q_{t-1} q_t = i, \mathbf{o}_1 \mathbf{o}_2 \cdots \mathbf{o}_t | \lambda], \tag{4.45}$$

as the highest probability along a single path, at time t, which accounts for the first t observations and ends in state i. Based on the above definition, the complete procedure for finding the best state sequence is given as:

1. Initialization

$$\delta_1(i) = \pi_i b_i(\mathbf{o}_1), \qquad i = 1, \cdots, N \tag{4.46}$$

$$\psi_1(i) = 0. \tag{4.47}$$

2. Recursion

$$\delta_t(j) = \max_{1 \le i \le N} [\delta_{t-1}(i) a_{ij}] b_j(\mathbf{o}_t), \qquad \begin{array}{c} t = 2, \cdots, T \\ j = 1, \cdots, N \end{array} \tag{4.48}$$

$$\psi_t(j) = \arg \max_{1 \le i \le N} [\delta_{t-1}(i) a_{ij}], \qquad \begin{array}{c} t = 2, \cdots, T \\ j = 1, \cdots, N. \end{array} \tag{4.49}$$

3. Termination

$$P^* = \max_{1 \le i \le N} [\delta_T(i)] \tag{4.50}$$

$$q_T^* = \arg \max_{1 \le i \le N} [\delta_T(i)]. \tag{4.51}$$

4. Path (state sequence) backtracking

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \qquad t = T-1, T-2, \cdots, 1. \tag{4.52}$$

The probability of this single best path is the matching score for the given input against the HMM for the claimed identity and is given by Equation (4.50) above. However, on most machines a direct computation of the above probability will result in numerical underflow. Hence, it is customary to calculate the matching score as the logarithm of the probability. Thus, for a given digit the matching score against the speaker's model (S) is given as

$$L_i(S) = \log(P_i^*(S)), \qquad i = 1, 2, 7, 9. \tag{4.53}$$

In our system the input consists of one utterance containing each of the four digits. Hence, the probability of the utterance being spoken by the user is the product of the individual probabilities under the assumption that the digits were spoken in-

dependently. However, since we are dealing with the logarithm of the probability, the combined matching score can be simply computed by adding together the matching score for each utterance against its corresponding model:

$$L(S) = L_1(S) + L_2(S) + L_7(S) + L_9(S). \tag{4.54}$$

Recent studies [42, 43, 44] have suggested that the log likelihood score defined above can be normalized by the use of the so called *cohort models*. The test utterance is scored against the cohort models and the resulting score is subtracted from the log likelihood score defined above. The normalized matching score thus obtained has been demonstrated to be more stable and less variable than the unnormalized score and thus results in a better performance.

Several cohort models have been proposed [43]. However, in our system we make use of the speaker-independent models that are used for speech recognition as the cohort models. Thus cohort normalization is done by computing the combined matching score for the test utterance against the speaker independent models and subtracting that score from the combined matching score given by Equation (4.54) above.

4.4 Experimental Results in Literature

The YOHO database [45] has been widely used for evaluating the performance of speaker verification systems. The database consists of combination lock sequences

(e.g., 26-47-94) collected from 138 individuals over a period of three months in an office environment using a STU-III electret-microphone telephone handset. Each individual has four enrollment sessions with 24 phrases per session and 10 verification sessions with four phrases per session. For the YOHO database, Che and Lin [46] report an equal error rate of 0.62% using cepstrum features and word-level HMM models. For the same database, Colombi *et al.* [44] report an equal error rate of 0.28% using a feature vector composed of the cepstrum, log frame energy, first-order delta cepstrum and second-order delta cepstrum. However, they use phoneme-level HMM models rather than word-level HMM models and they also employ cohort normalization.

It should, however, be noted that in-spite of the impressive performance figures stated above, speaker verification systems have not been able to achieve comparable accuracies in field tests.

4.5 Summary

The production of human speech is a complex process that involves several transformations occurring at different levels: semantic, linguistic, articulatory, and acoustic [33]. Anatomical variations that occur among different people and the differences in their learned speaking habits manifest themselves as differences in the acoustic properties of the speech signal. By analyzing and identifying these differences, it is possible to discriminate among speakers [33]. The verification process, using speech, can be text-dependent or text-independent.

We have implemented a text-dependent speaker verification system. Our vocabu-

lary consists of four digits: *one, two, seven* and *nine*. Hidden Markov models are used to model the speech production process for each digit and the matching score is the probability that a given speech sample was generated by the HMM for the claimed identity. Speech recognition is also implemented to ensure the validity of the input speech and for use in implementing cohort normalization.

Speech is probably the most readily accepted biometric and, in fact, the largest scale deployment of any biometric to date is Sprint's Voice FONCARD [33]. The reason for this popularity is simple: acquisition of speech samples is a very simple non-invasive procedure. Furthermore, speech acquisition requires very modest hardware and, in most cases, it is possible to perform identity authentication over the existing telephone networks. However, compared to fingerprints or retinal scans, the discriminative capability of a voice-print is significantly lower. Consequently, the performance of a speaker verification system for a large population is limited. Furthermore, environmental factors such as background noise, the time of the day, the transmission channel, as well as the emotional state of the individual, cause a significant change in speech patterns. Also people skilled in mimicking other peoples voices may be able to easily fool a speaker verification system. Consequently, it is generally agreed that a speaker verification system will not be able to achieve an accuracy comparable to fingerprint-based or retinal pattern-based biometric systems.

Chapter 5

Decision Fusion

Each of the three biometric systems considered so far makes use of only one biometric characteristic to compute a confidence about the identity claim made by an individual. An integration schema that combines the confidence levels expressed by the independent biometric systems is likely to improve the verification accuracy. In this chapter we propose two different decision fusion schemas that allow us to integrate face, fingerprint and speech to improve the accuracy of the system in terms of FAR and FRR.

Identity authentication using multiple biometric indicators is essentially a decision fusion problem, which utilizes information from multiple systems to increase the fault-tolerance, to reduce uncertainty, to reduce noise, and to overcome the limitations of individual systems [47]. A multi-modal approach can increase the reliability of the decisions made by a biometric system. Multiple biometrics enable a user's identity to be verified even if some of the biometric characteristics used by the system are not available and/or not suitable for automatic processing. By using multiple

biometric characteristics, the system will be applicable to a larger target population. In addition, a multi-modal biometric system is generally more robust to fraudulent technologies, because it is more difficult to forge multiple biometric characteristics than to forge a single biometric characteristic. Figure 5.1 depicts the architecture of a generic biometric-based identity verification system employing multiple biometrics [4].

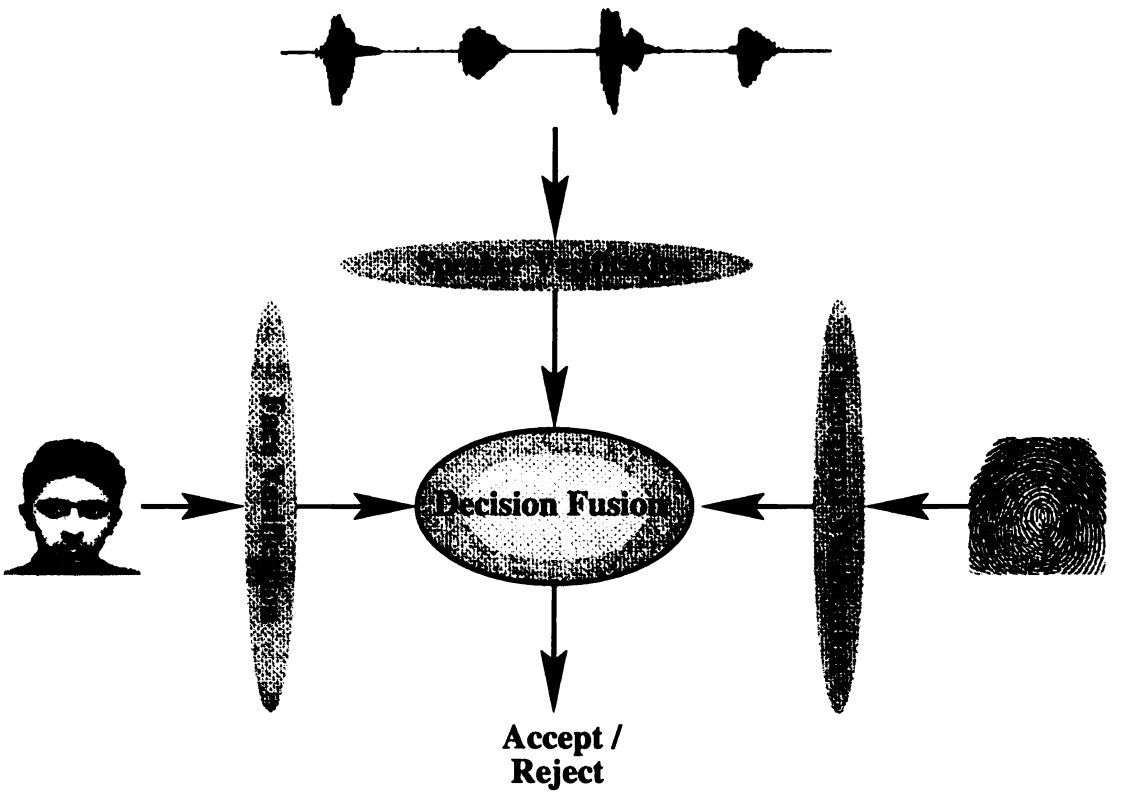


Figure 5.1: Integration of different biometric characteristics (after Jain *et al.* [4]).

Decision fusion can be carried out at different levels [5] : (*i*) Abstract level; the output from each module is only a YES/NO label without any confidence associated with the labels; in this case, the simple majority rule may be employed to reach a more reliable decision [48], (*ii*) Rank level; the output from each module is a YES/NO

label and the modules are ranked by decreasing confidence values, but the confidence values themselves are not specified; (*iii*) Measurement level; the output from each modules is a YES/NO label with associated confidence values; in this case, more accurate decisions can be made by integrating different confidence measures to a more informative confidence measure. Each of the biometric systems considered so far has a very different characteristic and a different matching scheme. Therefore, it is more reasonable to integrate the three biometrics at the measurement level instead of at the other two levels. An identity claim along with the requisite biometrics are inputted to the integrated system, the different modules within the system process the respective biometric and produce an accept/reject decision along with a score indicating the confidence in their decision. The decision fusion module then makes a final accept/reject decision.

5.1 Formulation

Let $\mathcal{B}$ denote a given biometric system, and let $\Phi^1, \Phi^2, \cdots, \Phi^N$ denote the templates of the N users enrolled in $\mathcal{B}$, who are labeled by numeric indicators, $1, 2, \cdots, N$ [4]. Assume, for simplicity, that each enrolled user has only one template (for each type of indicator) stored in the system. So the template for the ith user, $\Phi^i = \{\Phi_1^i, \Phi_2^i, \Phi_3^i\}$, has three components, where $\Phi_1^i, \Phi_2^i, \Phi_3^i$ are the templates for fingerprint, face, and speech biometrics, respectively. Let (Φ^0, I) denote the biometric indicator and the identity claimed by a user. Again Φ^0 has three components, $\Phi^0 = \{\Phi_1^0, \Phi_2^0, \Phi_3^0\}$, corresponding to the measurements of the three biometric indicators. The claimed

identity, I, belongs to either category w_1 or category w_2, where w_1 indicates that the user claims a true identity (a genuine user) and w_2 indicates that the user claims a false identity (an impostor). The biometric system $\mathcal{B}$ matches Φ^0 against Φ^I to determine into which category, w_1 or w_2, the claimed identity I falls, *i.e.*

$$I \in \begin{cases} w_1, & \text{if } \mathcal{F}(\Phi^0, \Phi^I) > \epsilon, \\ w_2, & \text{otherwise,} \end{cases} \tag{5.1}$$

where $\mathcal{F}(\Phi^0, \Phi^I)$ is a function which measures the similarity between Φ^0 and Φ^I and ϵ is a threshold. Under the assumption that the three biometric indicators are independent, we can rewrite the function $\mathcal{F}(\Phi^0, \Phi^I)$ as follows:

$$\mathcal{F}(\Phi^0, \Phi^I) = \mathcal{F}(\mathcal{F}_1(\Phi_1^0, \Phi_1^I), \mathcal{F}_2(\Phi_2^0, \Phi_2^I), \mathcal{F}_3(\Phi_3^0, \Phi_3^I)), \tag{5.2}$$

where $\mathcal{F}_1(\Phi_1^0, \Phi_1^I)$, $\mathcal{F}_2(\Phi_2^0, \Phi_2^I)$, and $\mathcal{F}_3(\Phi_3^0, \Phi_3^I)$ are functions that measure the similarity between the corresponding biometric indicators. For the three biometrics considered so far, we have:

1. Fingerprints: Let Φ_1^I denote the extracted minutiae for the Ith identity and Φ_1^0 denote the extracted minutiae for the input fingerprint image. Then

$$\mathcal{F}_1(\Phi_1^0, \Phi_1^I) = \frac{100C^2}{PQ}, \tag{5.3}$$

where P and Q are the total number of minutiae in Φ_1^0 and Φ_1^I, respectively, and C is the total number of corresponding minutiae pairs between Φ_1^0 and Φ_1^I established by the minutiae matching algorithm.

2. Face: Let Φ_2^I denote the projection of the face image for the Ith identity onto the eigenspace and let Φ_2^0 denote the projection of the input face image into the same eigenspace. Then

$$\mathcal{F}_2(\Phi_2^0, \Phi_2^I) = -||\Phi_2^I - \Phi_2^0||, \tag{5.4}$$

where $||\cdot||$ denotes the L_2 norm.

3. Speech: Let Φ_3^I denote the hidden Markov model for a given word for the Ith identity and $\Phi_3^0 = (\mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_T)$ denote the sequence of feature vectors extracted from the input speech. Then

$$\mathcal{F}_3(\Phi_3^0, \Phi_3^I) = P((\mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_T)|\Phi_3^I). \tag{5.5}$$

As explained in chapter 4, the probability computation step is performed using the Viterbi algorithm (i.e., the maximum likelihood path is used).

When the input biometric indicator belongs to a genuine individual, the resulting similarity measure is called a *genuine* score and when it belongs to an impostor it is called a *impostor* score. If we denote the three similarity measures defined above by the random variables X_1, X_2, and X_3 then the distributions of genuine score for fingerprint, face and speech verification are given by the class-conditional probability functions $p_j(X_j|w_1)$, where $j = 1, 2, 3$. Similarly, the distributions of impostor score are given by the class-conditional probability functions $p_j(X_j|w_2)$, where $j = 1, 2, 3$. Further, under the assumption that X_1, X_2, and X_3 are statistically independent, the joint class-conditional probability density function of X_1, X_2, and X_3 has the following form:

$$p(X_1, X_2, X_3|w_i) = \prod_{j=1}^{3} p_j(X_j|w_i), \qquad i = 1, 2. \tag{5.6}$$

5.2 The Neyman-Pearson Rule

Depending on the application requirement on verification accuracy, any one of a number of different statistical decision theory frameworks can be used. In biometrics, the performance requirement is usually specified in terms of the FAR [1]. In this case, the decision fusion scheme should establish a decision boundary which satisfies the FAR specification and minimizes the FRR. Let R^3 denote the three-dimensional space

spanned by (X_1, X_2, X_3); R_1^3 and R_2^3 denote the w_1-region and w_2-region, respectively ($R_1^3 + R_2^3 = R^3$); ϵ_0 denote the specified FAR. According to the Neyman-Pearson rule[49], a given observation, $X^0 = (X_1^0, X_2^0, X_3^0)$, is classified as:

$$(X_1^0, X_2^0, X_3^0) \in \begin{cases} w_1, & \frac{p(X_1^0, X_2^0, X_3^0|w_1)}{p(X_1^0, X_2^0, X_3^0|w_2)} > \lambda \\ w_2, & \text{otherwise} \end{cases} \tag{5.7}$$

where

$$\lambda = \frac{p(X_1, X_2, X_3|w_1)}{p(X_1, X_2, X_3|w_2)} \quad \text{and} \tag{5.8}$$

$$\epsilon_0 = \int_{R_1} p(X_1, X_2, X_3|w_2) dX_1 dX_2 dX_3. \tag{5.9}$$

For a given biometric system, the genuine and the impostor class-conditional probability density functions are usually unknown. A critical issue in this decision fusion scheme is to estimate the genuine and impostor class-conditional probability density functions from a set of training samples. Ideally, it would be desirable to be able to characterize these probability density functions by known statistical models. However, in practice, these genuine and impostor probability density functions are estimated using a non-parametric technique from empirical data.

5.3 Linear Discriminant Functions

In addition to the statistical decision theory approach, decision fusion can also be performed using a discriminant function approach:

$$(X_1^0, X_2^0, X_3^0) \in \begin{cases} w_1, & \text{if } \mathcal{F}(X_1^0, X_2^0, X_3^0) > a_0 \\ w_2, & \text{otherwise,} \end{cases} \tag{5.10}$$

where $\mathcal{F}$ is the discriminant function and a_0 is a threshold, which can be "learned" from a set of training samples using a number of techniques. While the discriminant function $\mathcal{F}$ can be of any form, in practice, a linear function is used for its simplicity:

$$(X_1^0, X_2^0, X_3^0) \in \begin{cases} w_1, & \text{if } (a_1 X_1^0 + a_2 X_2^0 + a_3 X_3^0) > a_0 \\ w_2, & \text{otherwise,} \end{cases} \tag{5.11}$$

where a_0, a_1, a_2, and a_3 are unknown parameters which need to be estimated from training data. An important problem in this approach is to find an appropriate discriminant function that satisfies the specified FAR. Generally, there exists no systematic method to find such a discriminant function. It is, however, possible to use a perceptron to obtain a decision surface that minimizes the equal error rate and then the desired decision scheme can be established by moving the resulting decision surface parallel to itself so that the FRR is minimized for the specified FAR.

5.4 Summary

Due to the inherent limitations of each of the biometric characteristic (refer to Table 1.1) it is difficult for an automatic personal identity authentication system based on only a single biometric characteristic to achieve the desired verification accuracy. A multimodal biometric system that integrates more than one biometric characteristic through a decision fusion scheme can overcome the limitations of the individual biometric characteristic. Thus, if the person is unable to give a good fingerprint sample due to various reasons (dryness of the skin, personal injury, etc.) then the system may use only his/her face and speech samples for identity authentication. Similarly, in a noisy environment, a biometric system based on speech alone may not be able to reach a reliable decision, while in a cluttered environment face verification may not work very well. We have proposed two decision fusion schemes that integrate faces, fingerprints and speech so as to improve the verification accuracy. The proposed schemes also make the integrated system applicable to a larger target population and diverse operating environments.

Decision fusion can also be employed for automatic personal identification systems wherein the goal is to improve the accuracy as well as the speed of the integrated system [8]. It can also be employed in systems wherein a single biometric characteristic is processed by multiple matching algorithms.

Chapter 6

Integrated Biometric System

Our goal is to design and implement a fully automatic personal verification system that integrates three biometrics: faces, fingerprint and speech. The system must be able to operate in real time and must be user-friendly. By real time we imply that the response time of the system must be of the order of a few seconds. "User-friendliness" in this context implies that the system must be simple to use and easy to maintain. Furthermore, the proposed system is intended for use in a *medium* sized environment wherein the number of enrolled users is of the order of a few thousands. The block diagram of such an integrated system is depicted in Figure 6.1. The system mainly consists of four components: (*i*) data acquisition module, (*ii*) enrollment module, (*iii*) template database, and (*iv*) verification module.

The data acquisition module is responsible for acquiring face and fingerprint images and speech samples of a user who intends to use the system. The enrollment module is then responsible for converting them into a format suitable for use by the verification module and storing them in the template database. In addition, the en-

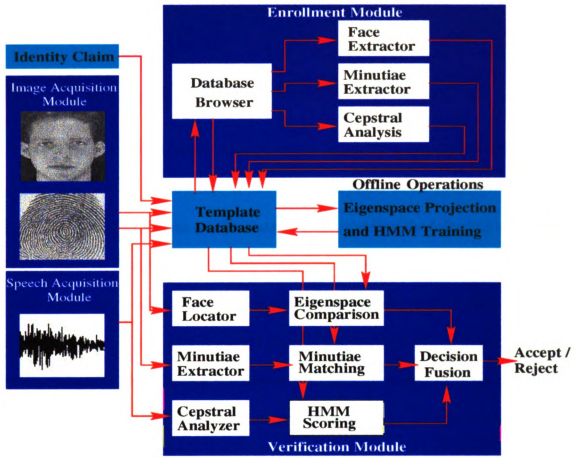


Figure 6.1: The block diagram of the integrated system for personal verification.

rollment module also allows for system maintenance operations such as user deletion, user update, system parameter specification, *etc.* The use of a template database allows for efficient retrieval of templates for a given identity claim during verification and also allows for any off-line operations that may be required. The verification module receives a set of input data from the data acquisition module and the identity claimed by the user. It then performs feature extraction on the input data and retrieves the representative templates for the claimed identity. Finally, it evaluates the input data against the templates and makes use of decision fusion to reach an

accept/reject decision.

6.1 Data Acquisition Module

The performance of a biometric-based personal verification system depends critically on the data acquisition process. If the user is assumed to be cooperative then the quality of the acquired data can be controlled by providing the following mechanisms to the user: (*i*) feedback and self-regulation guide, and (*ii*) quality control. For a cooperative user, feedback provides an efficient mechanism to guarantee the quality of input images; if the user does not provide the corresponding biometric characteristic properly, the feedback mechanism will allow the user to make appropriate adjustments. We have designed an intuitive graphical user interface to acquire face and fingerprint images and speech data (Figure 6.2). For face and fingerprint images, real-time feedback allows the user to rapidly adjust the location of the finger or the position of the head so as to satisfy the placement requirements for face and fingerprint images. For speech data, the quality of the input speech in terms of signal to noise ratio, *etc.* is monitored and feedback is provided to the user accordingly.

6.2 Enrollment Module

In our system, the data acquisition process is differentiated from the enrollment process so as to allow for data collection to be done in a distributed fashion. The enrollment module allows for centralized system administration. Once the data is

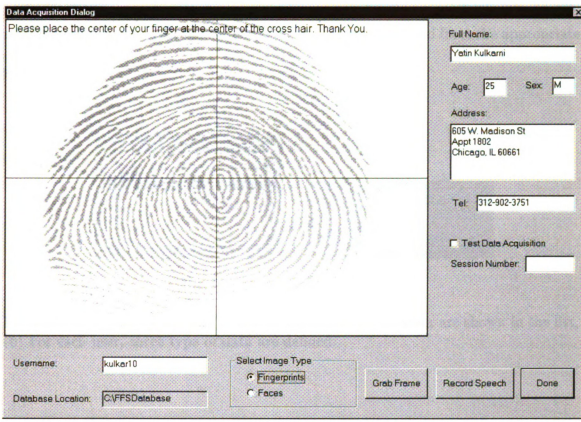
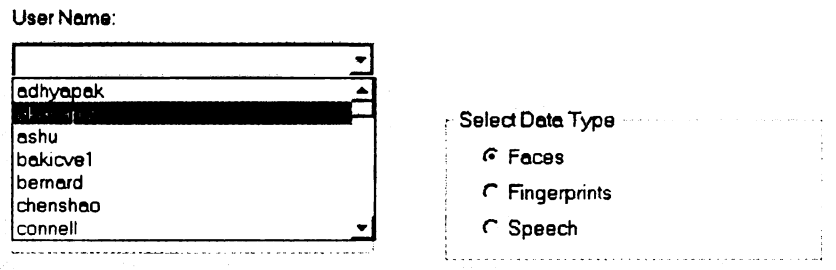


Figure 6.2: GUIs for data acquisition during enrollment.

collected for a particular user, the following events must occur before the user is considered to be enrolled in the system: (i) the eigenspace and the projections onto the eigenspace of the user must be computed, (ii) a representative image from the given set of fingerprint images must be selected and minutiae extraction must be applied, and (iii) hidden Markov models must be trained for the user.

The graphical user interface for the enrollment module has been designed to facilitate the above three tasks for an administrator. The administrator can select a user from the list of users in the database (Figure 6.3(a)). For the selected user, the particular data to be browsed can be selected from the three possible choices (Figure

6.3(b)). The GUI changes to adapt itself to the data type selected (Figure 6.4). The required tasks to complete the enrollment can now be performed from the appropriate screens.



(a) (b)

Figure 6.3: User Database: (a) The users enrolled in the system are shown in the list, (b) For each user, three type of data are defined.

6.3 Template Database

Ideally, a relational database management system would be used for storing the templates generated for each user. However, since our system is intended for use in a medium sized environment, it suffices to use the native file system for the given hardware and software platform for storing the templates. The template database is organized as a hierarchical directory structure which is indexed by the user name. For each user, the face template consists of the projections of the face images onto the eigenspace defined by the set of enrolled users. The fingerprint template consists of the set of extracted minutiae from the selected fingerprint image (the selection is done manually in the enrollment module). The speech template consists of the

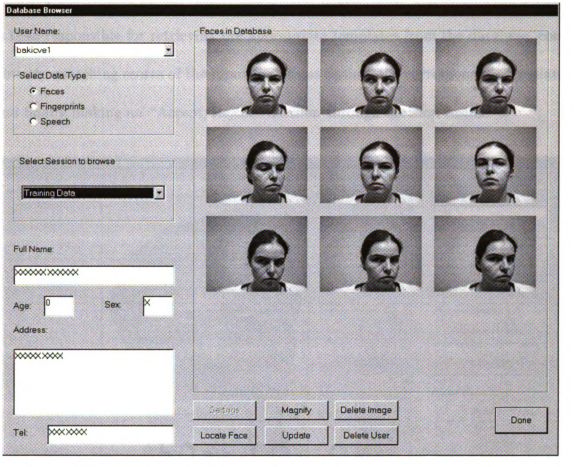


Figure 6.4: GUIs for the enrollment module.

four hidden Markov models for that user. The combined storage requirement for the three templates for a single user is of the order of a few thousand kilobytes (typically around 7 to 8 KB).

6.4 Verification Module

During the verification session, the user is required to give one rendition of each of the three biometrics and submit an identity claim. Since the data acquisition method at verification stage is different from the enrollment session, we have designed a different

data acquisition GUI for the verification module (Figure 6.5). The verification module is then responsible for retrieving the appropriate templates from the database, compute the matching scores of the input renditions against the corresponding templates and finally making an "Accept/Reject" decision through decision fusion.

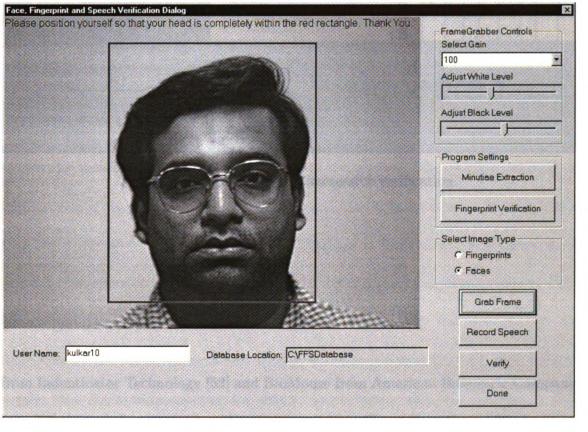


Figure 6.5: GUIs for the verification module.

6.5 Commercially Available Biometric Systems

Today a large number of vendors are offering automatic personal identification and verification systems in the market. Some of the well-known systems are TrueFace from Miros [50] and FaceIt from Visionics [51] that are face-based systems, GateKey

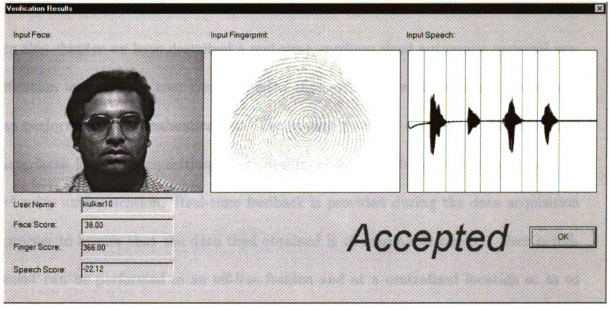


Figure 6.6: The result of a successful verification.

from Indenticator Technology [52] and BioMouse from American Biometric Company Inc. [53] that are fingerprint-based systems, and VoiceCrypt from Veritel Corporation of America [54], SpeakeEz from T-Netix [55] and SpeakerKey from ITT [56] that are speech-based systems. All the above systems are available for the Microsoft Windows platform. Admittedly, our system does not have the same commercial appeal as these systems but the functionality provided by our system is comparable to any of the above systems. Also, all the above systems make use of a single biometric. We are not aware of any commercially available system for the Microsoft Windows platform that uses multiple biometrics.

6.6 Summary

In this chapter we have described a real-time, medium sized automatic personal verification system that uses three different biometrics: face, fingerprint, and speech to perform identity authentication. The system provides convenient graphical user interfaces for data acquisition, user enrollment and database management and for identity authentication. Real-time feedback is provided during the data acquisition process to ensure that the data thus obtained is of reasonable quality. User enrollment can be performed in an off-line fashion and at a centralized location so as to ensure the validity of the users being added to the system. The verification module requires one instance each of face, fingerprint and speech along with an identity claim to perform authentication. The template database is maintained as a hierarchical directory structure on the native file system of the host machine, in our case, an IBM compatible personal computer running Windows NT 4.0. In the next chapter we will evaluate the performance of our system and demonstrate through experimental results that the system achieves the desired performance and also operates in real time.

Chapter 7

Experimental Results

In this chapter we shall attempt to evaluate the performance of our integrated system and assess the gain in performance that can be achieved by employing the two decision fusion schemes discussed in Chapter 5. We will demonstrate that our proposed decision fusion schemes significantly improve the performance of the integrated system over the individual biometric systems. It is, however, critical to understand that the performance of a biometric system is dependent upon the operating environment and the amount of cooperation that can be expected from the user. Hence, we must first understand the various factors that affect the performance of the biometric systems based on face, fingerprint, and speech. These factors are summarized in Table 7.1. The users are assumed to be cooperative. *Controllability* refers to the ease with which the various factors can be constrained to be within given limits for a cooperative user without causing undue inconvenience to the user.

The performance evaluation of a biometric system is a difficult task [57, 58]. Given any biometric system, it is extremely time consuming and expensive for the

Biometric	Factors	Controllability
Face	Illumination	High
	Distance	High
	Orientation	Medium
	Cosmetics	Low
	Facial Expressions	Low
Fingerprint	Illumination	High
	Impression pressure	Medium
	Orientation	Medium
Speech	Ambient Noise	Medium
	Spectral Variations	Low
	Device Variations	Medium

Table 7.1: Various factors affecting the performance of different biometric systems.

manufacturer to test it under all possible conditions under which the system may have to operate. As of today, there are no standard benchmarks available for a biometric system. The performance of a biometric system can be quantified by various measures of performance such as FAR, FRR, receiver operating curve, etc. However, these performance metrics are all dependent on the database available for training and testing and the conditions under which the data was collected. Hence proper care needs to be taken while generating the database. The number of individuals in the database should be large enough so as to adequately represent the population and enough samples must be available for each individual [57]. If these conditions are met then the resulting performance metric *can* be a fair estimate of the true performance of the system when used with the entire population.

We shall make use of the receiver operating characteristics (ROC) to assess the performance of our system. The ROC curve is a plot of the *authentic acceptance rate* $(1 - FRR)$ against the FAR. The ROC curve is obtained by varying the operating

point of the system from a point where the FRR is 100% to a point where the FAR is 100%. Ideally, we would like to have a system with a 100% authentic acceptance rate for all values of FAR. This would imply that a genuine individual is never rejected. The ROC curve for such a system is depicted in Figure 7.1. In practice, due to the inherent variability in the data acquisition process, it is impossible for a system to have such a ROC curve. The objective is then to design a system with a ROC curve that is as close as possible to the ROC curve depicted in Figure 7.1.

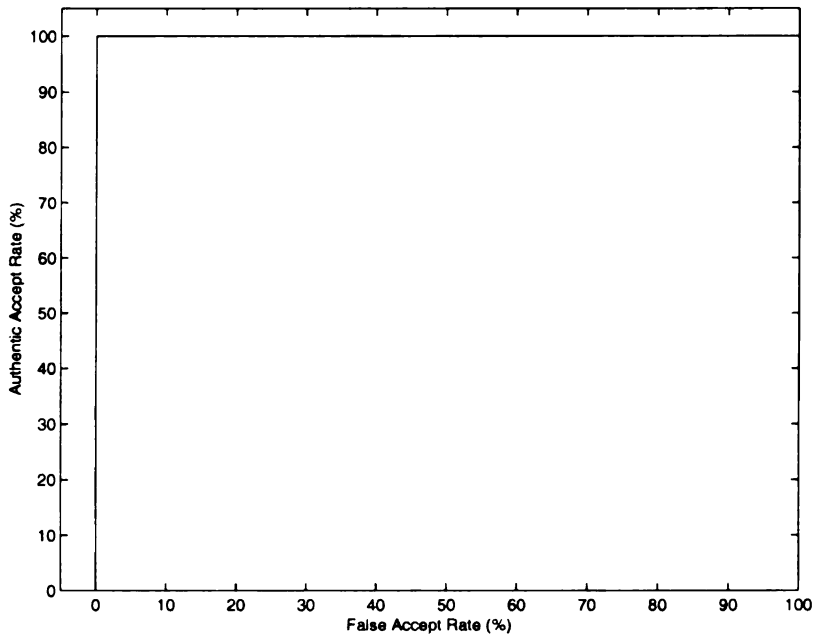


Figure 7.1: An ideal receiver operating characteristics curve.

7.1 Database

We have collected a database of faces, fingerprints and speech samples from 50 individuals. For each of these 50 individuals, we have collected 9 samples each of their frontal facial images, 10 samples each of their fingerprints from a single finger and

24 samples each of their speech. The frontal facial images were obtained using a Panasonic camera under adequate light and against a plain background. The images were digitized using a frame-grabber from Data Translation and were re-sampled and cropped from their original size of 640 × 480 to the desired size of 80 × 90. Finally, the images were normalized to have zero mean. The subjects were asked to vary the orientation of the head and the direction of their gaze within small amounts across each of the nine samples. A typical set of face images obtained for an individual are depicted in Figure 7.2.

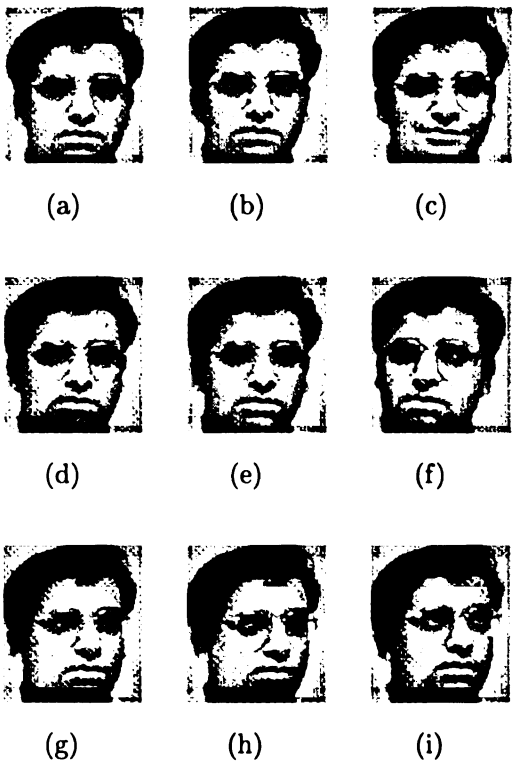


Figure 7.2: A typical example of the nine face images acquired for each user.

The fingerprint images were obtained using a live-scan fingerprint scanner manufactured by Digital Biometrics. The fingerprint acquisition process was supervised in

terms of controlling the position, orientation and quality of the fingerprints. Consequently, all of the acquired fingerprints are of reasonable quality. Figure 7.3 depicts typical fingerprints in the database.

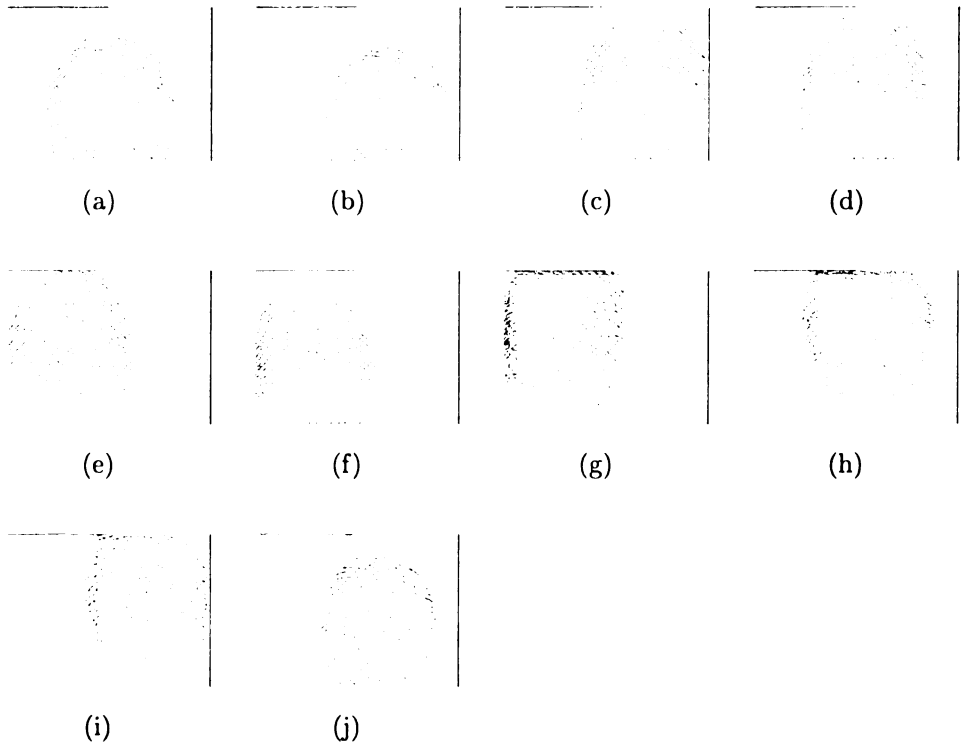


Figure 7.3: A typical example of the ten fingerprint images of good quality acquired for each user.

Our speech database consists of 24 samples per speaker, two sets of 12 samples each, collected over two sessions held approximately a week apart. The recordings were done in a laboratory environment using a Labtec microphone employing noise reduction technology and a Sound Blaster audio card. The resulting speech samples have a very high signal-to-noise ratio. Figure 7.4 depicts a typical speech sample consisting of the four digits one, two, seven and nine.

The database described above was used for generating the face, fingerprint and

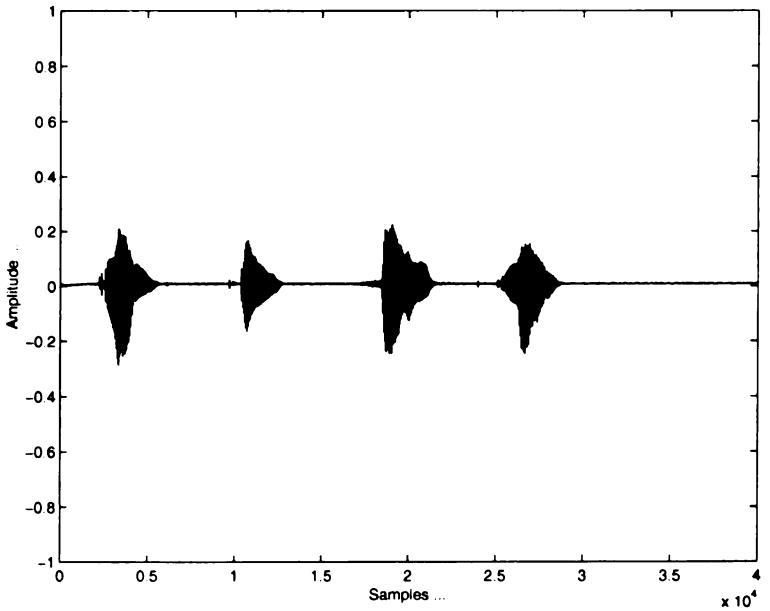


Figure 7.4: The waveform of a typical speech sample in our database.

speech templates for the users and for training the decision fusion schemes. Next, for a group of 25 people, additional sets of data were collected, over multiple sessions. In each such session, 5 samples each of face, fingerprint and speech were collected. On an average, about 3 such sessions were conducted for each of the 25 users. Finally, a database of impostors was generated by pooling together face, fingerprint and speech samples of individuals not enrolled in the system. The performance of the system was also evaluated on this database of impostors. The genuine and the impostor databases were pooled together to evaluate the performance of the individual and the integrated systems.

7.2 System Training

The first step in the evaluation process is to generate the impostor and genuine distributions for face, fingerprint and speech matching scores. We shall briefly discuss the methodologies used for generating the impostor and genuine distributions for the individual biometrics. We note that the matching scores for face, fingerprint and speech will have their respective ranges. Hence, we normalize the genuine and impostor distributions for each biometric to the range $[0, 100]$.

Face

In order to generate the genuine and impostor matching scores, the leave-one-out method was employed. Thus a face template for a user is obtained by using eight of the nine facial images. The image left out is used for computing the genuine and impostor matching scores. The process is then repeated for each of the nine images. Thus, for a database of 50 individuals with 9 face images per individual, the comparison of each image with the 50 templates results in one genuine matching score and 49 impostor matching scores. There are a total of 450 facial images, resulting in a total of 450 genuine scores and 22,050 impostor scores. The genuine and impostor distributions for face matching are depicted in Figure 7.5.

Fingerprint

Out of the ten fingerprints available for each user, we have manually selected one fingerprint as the template. The manual selection of the template fingerprint is a

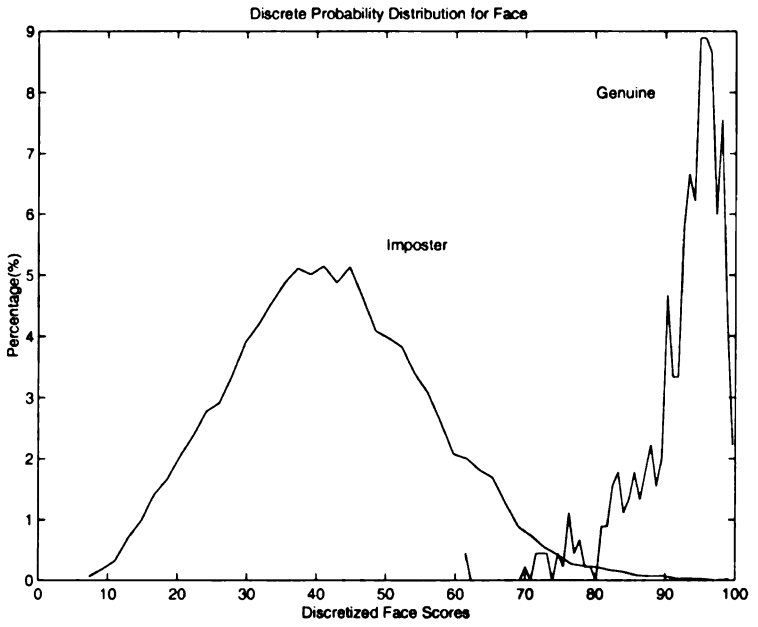


Figure 7.5: The genuine and impostor distributions for face matching.

reasonable choice so as to enable us to represent each individual using the most expressive sample. The remaining nine fingerprints are used to generate the impostor and the genuine distributions for the matching scores in a manner similar to that described for the face verification system. The genuine and impostor distributions for fingerprint matching are depicted in Figure 7.6.

Speech

Amongst the 24 samples for each user, 12 samples from one session were used for generating the hidden Markov models for that user and the 12 samples from the second session were used for generating the genuine and impostor distributions in a manner similar to the face verification system. The genuine and impostor distributions for speaker verification are depicted in Figure 7.7.

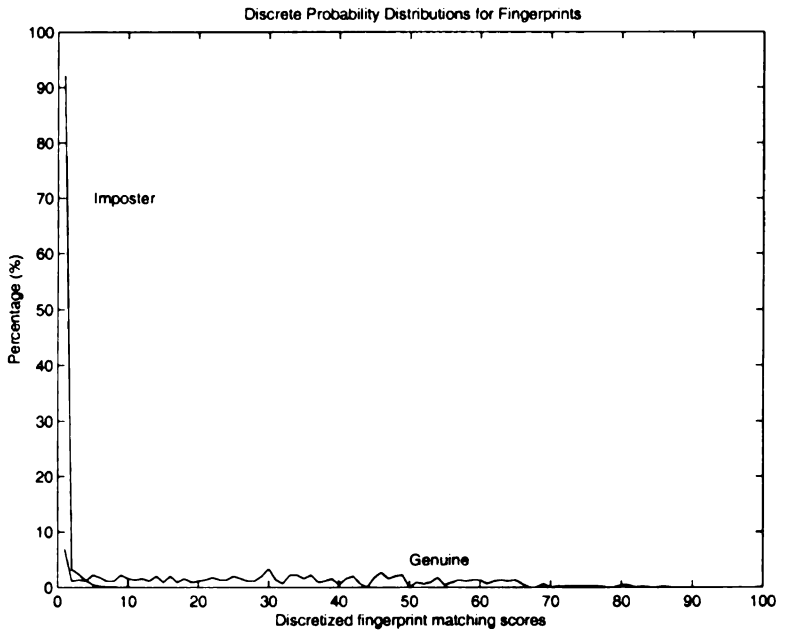


Figure 7.6: The genuine and impostor distributions for fingerprint matching.

The above distributions were then used to train the two decision fusion schemes discussed in Chapter 5 in the following fashion.

Neyman-Pearson Rule

The training procedure for the Neyman-Pearson rule essentially involves an exhaustive search in the $100 \times 100 \times 100$ dimensional space defined by the three discrete probability distributions. The objective is to define a region wherein the false acceptance rate is less than the desired value and then define a test on the likelihood ratio, λ. The Neyman-Pearson lemma[49] guarantees that the likelihood ratio test, as defined in Equation 5.7, with this λ will be the most powerful test resulting in the smallest possible false reject rate for the specified false accept rate. We simplify the exhaustive search into a linear search by sorting the likelihood ratios for the training data in

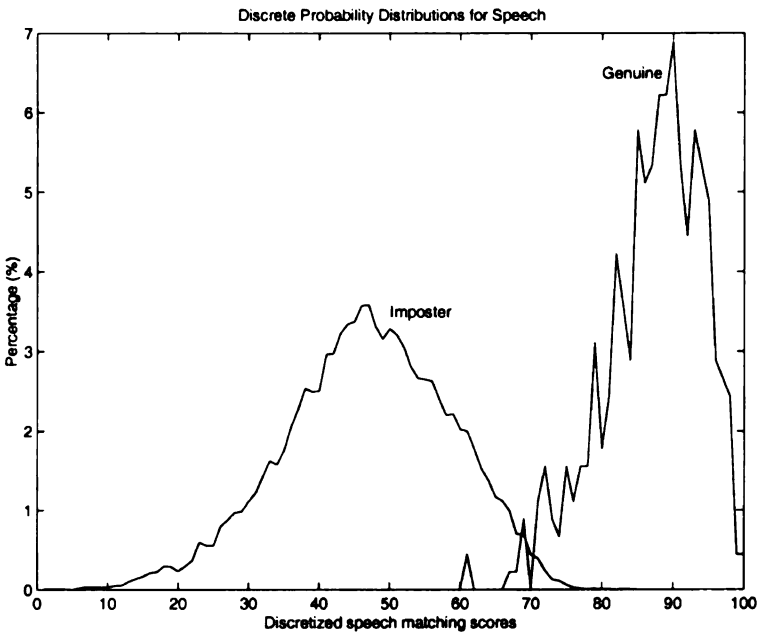


Figure 7.7: The genuine and impostor distributions for the speaker verification system.

an ascending order and then sequentially adding the sorted data points into the acceptance region without exceeding the predefined false accept rate.

Linear Discriminant Functions

Since the three impostor and genuine distributions are assumed to be independent of each other, the joint impostor and genuine distributions are simply obtained by combining the three univariate distributions in a three-dimensional Euclidean space. In this space the parameters of the linear discriminant function are learned using a perceptron. Since a perceptron can only be trained to minimize the equal error rate, a trial-and-error method is used next, wherein the decision surface obtained through perceptron training is moved parallel to itself till the desired FAR is achieved. Although, such an ad-hoc method is unlikely to be optimal, experimental results

demonstrate that the method works reasonably well.

7.3 Performance Evaluation

The purpose of this thesis is to demonstrate that the FAR and FRR of an automatic personal verification system can be improved by integrating multiple biometrics. Consequently, for each decision fusion schema, we first evaluate the performance of automatic personal verification systems based on each of the three biometrics by plotting the ROC curves for each system. Next we will evaluate the performance of the integrated system by plotting the ROC curve of the integrated system and demonstrate the gain in performance. The test data collected for the 25 individuals along with the impostor database was used to evaluate the various systems.

7.3.1 Performance of the Neyman-Pearson Decision Fusion Schema

The training procedure described above results in one likelihood ratio test for each desired value of the false accept rate. The test data is now used to evaluate the false accept rate and the false reject rate for each test on previously unseen data. The resulting receiver operating characteristics for the various systems are depicted in Figure 7.8.

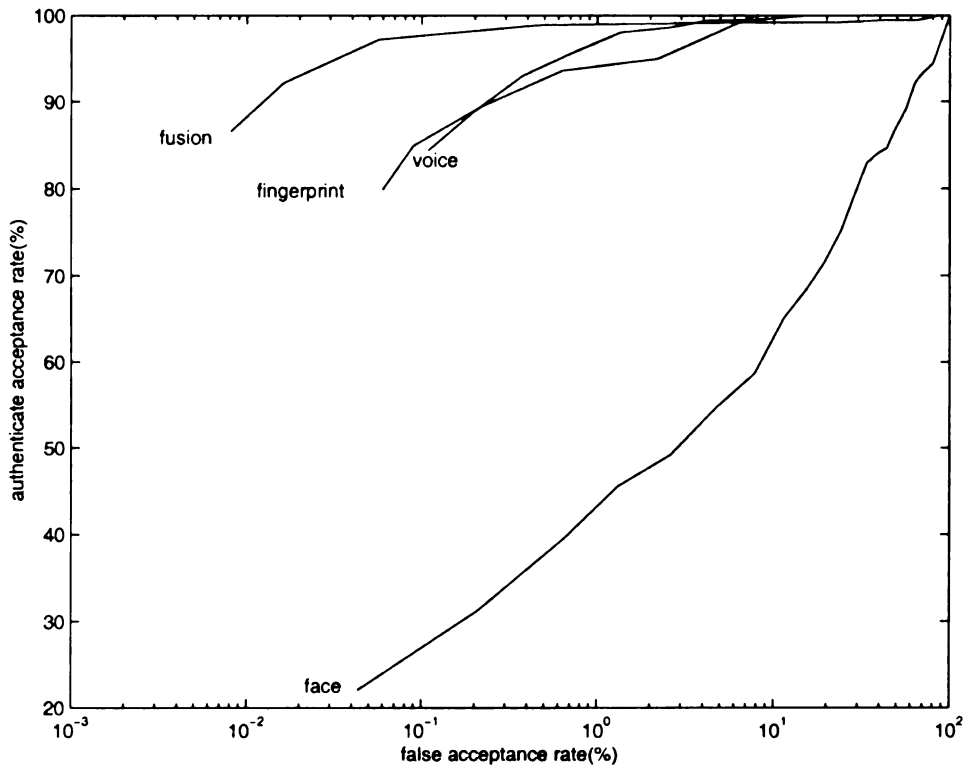


Figure 7.8: The receiver operating characteristics for the Neyman-Pearson rule.

7.3.2 Performance of the Linear Discriminant Function Decision Fusion Schema

The various linear discriminant functions obtained during the training phase are used along with the test data for performance evaluation. The resulting receiver operating characteristics for the various systems are depicted in Figure 7.9.

7.3.3 Verification Speed

Since the integrated system is intended to operate in real time, it is necessary that the response time of the system be of the order of a few seconds. The average wall-clock time required for our system for one verification session on a Pentium 200MHz

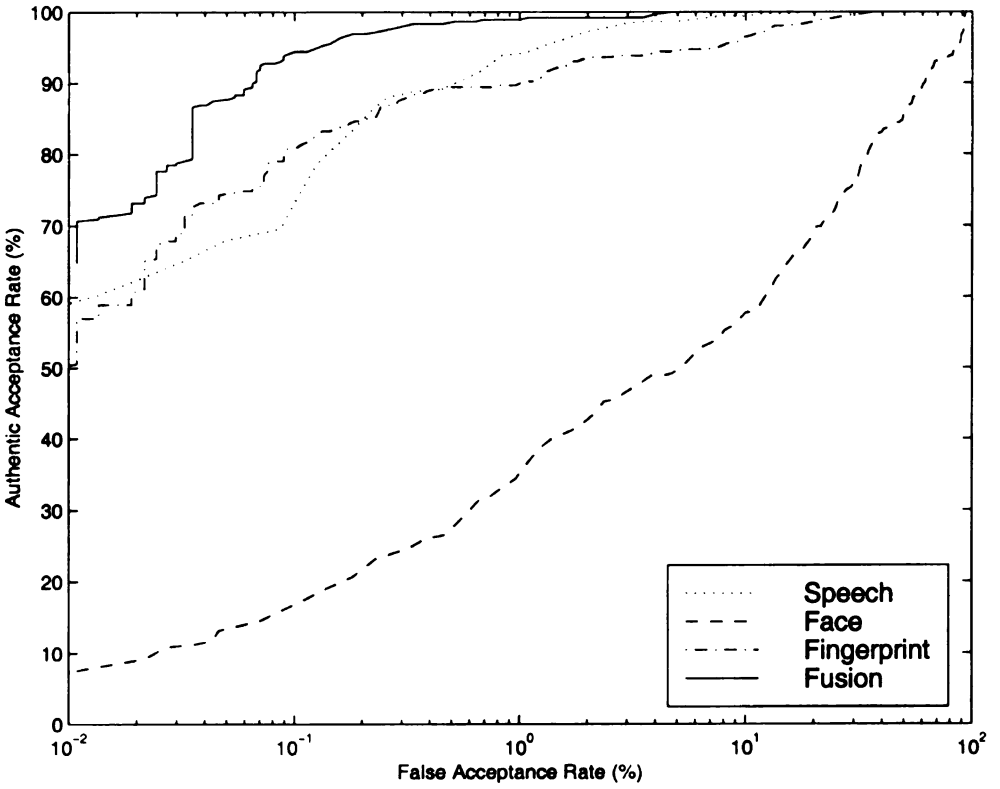


Figure 7.9: The receiver operating characteristics for the Linear Discriminant functions.

machine running Windows NT 4.0 is summarized in Table 7.2.

	Face	Fingerprint	Speech	Integrated System
Time (seconds)	0.5	1.5	0.75	3.0

Table 7.2: Wall-clock times for the various verification systems.

7.4 Summary

Obtaining a reliable estimate of the performance of a biometric system is a rather difficult task. To ensure that the estimated performance figures are reliable, the database used for evaluation must contain enough samples so as to adequately rep-

resent the intended population and the intended operating environment. Also, care must be taken while collecting the data to ensure that there is no bias towards or against any one of the three biometrics. The entire database is divided into three components: (*i*) data to be used for generating the templates, (*ii*) data to be used for training the various decision fusion schemes and (*iii*) data to be used for evaluating the generalization ability of the trained systems.

In this chapter we have presented the experimental results for the two proposed decision fusion schemes for a limited database collected in a laboratory environment and from cooperative users. For this database, we have demonstrated that the proposed decision fusion schemes are able to significantly improve the performance of the integrated system over the individual biometrics systems. The integrated system has a response time of approximately three seconds on a Pentium 200MHz machine running Windows NT 4.0. With code optimization and a faster processor, the algorithm is capable of operating in real-time.

Chapter 8

Summary and Future Research

In this chapter we will summarize the work we have done, discuss the limitations of our proposed approach and propose some directions for future research.

8.1 Summary

Biometrics, which is defined as the use of human physiological and behavioral characteristics to establish and/or verify the identity of an individual is poised to become the popular security standard in the near future. It is inherently more secure than knowledge-based or token-based security systems since it relies upon something that *one is* rather than something that *one knows* (e.g., passwords) or something that one has (e.g., magnetic stripe cards) to make an identification/verification. A number of different physiological and behavioral characteristics have been studied and identified as potential candidates for a biometric system. Amongst them face, fingerprint, and speech are the most widely applied and accepted biometric techniques. However,

each of these three biometrics has its limitations: face is the least accurate biometric; the performance of fingerprints is acceptable only when the fingerprint images obtained are of good quality; and the performance of speech depends heavily upon the operational environment and the data acquisition equipment. Thus, systems based on a single biometric are not able to achieve the performance required of a practical system. It is expected that a system that integrates these three biometrics can overcome the limitations of the individual biometrics and be suitable for practical use. Furthermore, such a system is also applicable to a larger target population.

In this thesis we have designed and implemented a real-time fully automatic personal identity verification system that integrates the following three biometrics: face, fingerprint, and speech. The system uses the eigenface method by Turk and Pentland [21] for face recognition/verification, the alignment-based minutiae matching algorithm by Jain *et al.* [2] for fingerprint verification and continuous mixture density hidden Markov models for speaker verification. A video camera is used to obtain frontal images of the individual's face under suitable lighting conditions and against a plain background. An optical fingerprint scanner is used to obtain the fingerprint images. The face and the fingerprint images are digitized using a frame grabber. The resulting images are of 640x480 size. Speech is recorded using a noise cancellation microphone and a standard sound card. The speech is sampled at a frequency of 8000 Hz and at a resolution of 16 bits per sample. The data thus obtained is then processed by the respective sub-systems resulting in three matching scores. These matching scores are then fed to a decision fusion module which uses one of the two proposed schemes to make the final "Accept/Reject" decision. Suitable graphical user

interfaces have been provided to make the verification system user-friendly and for providing real-time feedback during the data acquisition process.

Experimental results have been obtained on a limited database collected under a controlled laboratory environment and from cooperative users. These results demonstrate that our system achieves desired improvement in verification accuracy over the individual systems while maintaining the response-time requirements.

8.2 Future Research

In our system face verification can only be done under a controlled operating environment. This is mainly because of the lack of a reliable and fast face detection algorithm. The practicality of the system can be increased by incorporating a more sophisticated face detection algorithm.

The performance of the face verification system is the worst amongst the three biometrics. The primary reason for this is the elapsed duration between the training data collection and the test data collection. Figure 8.1 illustrates one instance of the difference between training and test data for the face system. Such changes are undesirable but unavoidable for a practical system.

Intra-class variations introduced by time affect the performance of all the three biometrics but to varying extent. Faces and speech are affected the most whereas fingerprints are affected only to a limited extent. It is, however, possible to overcome these variations by employing some kind of *template adaptation* technique. After each verification session, the test data is used to update the stored template. If the system

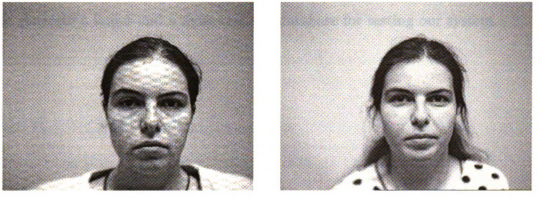


Figure 8.1: Example of the temporal variation present between training and test face images.

is used regularly then the temporal changes can be tracked and the performance can be maintained at the desired level.

Decision fusion is currently being performed at the measurement level. It is possible to perform decision fusion at a feature level. It is expected that a classifier designed to operate in the combined feature space is likely to be able to achieve a higher performance.

The integrated system is, at best, a useful tool for demonstration purposes. A number of enhancements need to be made to make it a commercially viable system. It would also be desirable to make a developer's toolkit available that could be used by application developers to easily integrate multimodal biometric authentication into their applications.

Finally, we note that the performance of our system has been tested on a limited database acquired in a laboratory environment. The performance on a larger database consisting of a few thousand users remains to be evaluated. We are currently in the

process of combining a number of different face, fingerprint and speech databases so as to generate a larger and a more realistic database for testing our system.

Bibliography

[1] E. Newham, "The biometric report," tech. rep., SJB Services, New York, 1995.

[2] A. K. Jain, L. Hong, S. Pankanti, and R. Bolle, "An identity-authentication system using fingerprints," *Proceedings of the IEEE*, vol. 85, pp. 1365 - 1388, September 1997.

[3] J. R. Deller, J. G. Proakis, and J. H. Hansen., *Discrete-time processing of speech signals.* New York: Macmillan Pub. Co., 1993.

[4] A. Jain, L. Hong, and Y. Kulkarni, "A multimodal verification system using faces, fingerprints and speech," in *To be published in the proceedings of AVBPA'99*, (Washington, D.C.), March 1999.

[5] R. Brunelli and D. Falavigna, "Person identification using multiple cues," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 955 - 966, October 1995.

[6] U. Dieckmann, P. Plankensteiner, and T. Wagner, "Sesam: A biometric person identification system using sensor fusion," *Pattern Recognition Letters*, vol. 18, pp. 827–833, September 1997.

[7] B. Duc, E. S. Bigun, J. Bigun, G. Maitre, and S. Fischer, "Fusion of audio and video information for multi modal person authentication," *Pattern Recognition Letters*, vol. 18, pp. 835–843, Spetember 1997.

[8] L. Hong and A. Jain, "Integrating faces and fingerprints for personal identification," *To appear in IEEE PAMI*, December 1998.

[9] S. Young, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK Book*. Cambridge, England: Entropic Cambridge Research Laboratories, 1997.

[10] Entropic, *Entropic Speech Technology*. http://www.entropic.com, 1998.

[11] A. Jain, L. Hong, and Y. Kulkarni, "F2id: A personal identification system using faces and fingerprints," in *Proceedings of the ICPR*, vol. 2, (Brisbane, Australia), pp. 1373–1375, August 1998.

[12] R. Chellappa, C. Wilson, and A. Sirohey, "Human and machine recognition of faces: A survey," *Proceedings of the IEEE*, vol. 83, pp. 705 – 740, May 1995.

[13] K. Sung and T. Poggio, "Example-based learning for view-based human face detection," A.I. Memo 1521, MIT A.I. Laboratory, 1994.

[14] T. K. Leung, M. C. Burl, and P. Perona, "Finding faces in cluttered scenes using random labelled graph matching," in *Proceedings of ICCV-'95*, (Cambridge, MA), 1995.

[15] H. Rowley, S. Baluja, and T. Kanade, "Nerual network-based face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 23–38, 1998.

[16] M. Lew and N. Huijsmans, "Information theory and face detection," in *Proceedings of ICPR-'96*, (Vienna, Austria), pp. 601–610, 1996.

[17] A. Colmenarez and T. Huang, "Face detection with information-based maximum discrimination," in *Proceedings of CVPR-'97*, (San Juan, Puerto Rico), 1997.

[18] K. C. Yow and R. Cipolla, "Feature-based human face detection," *Image and Vision Computing*, vol. 15, no. 9, pp. 713–735, 1997.

[19] S. H. Lin, S. Y. King, and L. J. Lin, "Face recognition/detection by probabilistic decision-based neural network," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 114–131, 1997.

[20] N. Duta and A. K. Jain, "Learning the human face concept from black and white pictures," in *Proceedings of the 14th ICPR*, vol. 2, (Brisbane, Australia), pp. 1365–1367, August 1998.

[21] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71 – 86, 1991.

[22] J. Zhang, Y. Yan, and M. Lades, "Face recognition: Eigenface, elastic matching and neural nets," *Proceedings of the IEEE*, vol. 85, pp. 1423 – 1435, September 1997.

[23] D. L. Swets and J. Weng, "Using discriminant eigenfeatures for image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 831–836, August 1996.

[24] R. Brunelli and T. Poggio, "Face recognition: Features versus templates," *IEEE Trans. PAMI*, vol. 15, no. 10, pp. 1042 – 1052, 1995.

[25] L. Hong, *Personal Identification Using Fingerprints.* PhD thesis, Department of Computer Science and Engineering, Michigan State University, 1998.

[26] N. Ansari, M. H. Chen, and E. S. H. Hou, *Dynamic, Genetic, and Chaotic Programming*, ch. A genetic algorithm for point pattern matching. John Wiley and Sons, 1992.

[27] H. Baird, *Model Based Image Matching Using Location.* Cambridge, MA: MIT Press, 1984.

[28] S. H. Chang, F. H. Cheng, W. H. Hsu, and G. Z. Wu, "Fast algorithm for point pattern matching: Invariant to translations, rotations, and scale changes," *Pattern Recognition*, vol. 30, no. 2, pp. 321–339, 1997.

[29] M. Eleccion, "Automatic fingerprint identification," *IEEE Spectrum*, vol. 10, no. 9, pp. 36–45, 1973.

[30] A. Ranade and A. Rosenfield, "Point pattern matching by relaxation," *Pattern Recognition*, vol. 12, no. 2, pp. 269–275, 1983.

[31] N. Ratha, K. Karu, S. Chen, and A. K. Jain, "A real-time matching system for large fingerprint database," *IEEE Transactions on PAMI*, vol. 18, no. 8, pp. 799–813, 1996.

[32] L. Rabiner and B. Juang, *Fundamentals of Speech Recognition.* Prentice Hall, 1993.

[33] J. P. Campbell, "Speaker recognition: A tutorial," *Proceedings of the IEEE*, vol. 85, pp. 1437 – 1462, September 1997.

[34] L. F. Lamel, L. R. Rabiner, A. E. Rosenberg, and J. G. Wilpon, "An improved endpoint detector for isolated word recognition," *IEEE Transactions on Accoustics, Speech, and Signal Processing*, vol. ASSP-29, pp. 777 – 785, August 1981.

[35] J. G. Wilpon, L. R. Rabiner, and T. Martin, "An improved word-detection algorithm for telephone-quality speech incorporating both syntactic and semantic constraints," *AT&T Bell Labs Technical Journal*, vol. 63, pp. 479 – 498, March 1984.

[36] S. Furui, "Cepstral analysis technique for automatic speaker verification," *IEEE Transactions on Accoustics, Speech, and Signal Processing*, vol. ASSP-29, pp. 254 – 272, April 1981.

[37] B. Atal, "Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification," *Journal of Acoustic Society of America*, vol. 55, no. 6, pp. 1304–1312, 1974.

[38] L. R. Rabiner, B. H. Juang, S. E. Levinson, and M. M. Sondhi, "Recognition of isolated digits using hidden markov models with continuous mixture densities," *AT & T Technical Journal*, vol. 64, pp. 1211 - 1234, July-August 1985.

[39] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, pp. 257 - 286, February 1989.

[40] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *J. Roy. Stat. Soc.*, vol. 39, no. 1, pp. 1–38, 1977.

[41] G. D. Forney, "The viterbi algorithm," *Proc. IEEE*, vol. 61, pp. 268–278, March 1973.

[42] A. E. Rosenberg, J. DeLong, C.-H. Lee, B.-H. Juang, and F. K. Soong, "The use of cohort normalised scores for speaker verification," in *Proc. ICSLP '92*, (Banff, Canada), Oct. 1992.

[43] A. E. Rosenberg and S. Parthasarathy, "Speaker background models for connected digit password speaker verification," in *Proc. ICASSP '96*, (Atlanta, GA), pp. 81–84, May 1996.

[44] J. Colombi, D. Ruck, S. Rogers, M. Oxley, and T. Anderson, "Cohort selection and word grammar effects for speaker recognition," in *Proc. ICASSP '96*, (Atlanta, GA), pp. 85–88, May 1996.

[45] J. J. P. Campbell, "Testing with the yoho cd-rom voice verification corpus," in *Proc. ICASSP '95*, (Detroit, MI), pp. 341–344, 1995.

[46] C. Che and Q. Lin, "Speaker recognition using hmm with experiments on the yoho database," in *Proc. EUROSPEECH*, (Madrid, Italy), pp. 625–628, 1995.

[47] A. K. Jain, R. Bolle, and S. Pankanti, eds., *Biometrics Personal Identification in Networked Society*, ch. 16. Kluwer Academic Publishers, 1998.

[48] Y. A. Zuev and S. K. Ivanov, "The voting as a way to increase the decision reliability," in *Proc. Foundations of Information/Decision Fusion with Applications to Engineering Problems.*, (Washington, D.C.), pp. 206–210, August 1996.

[49] T. Y. Young and T. W. Calvert, *Classification, Estimation and Pattern Recognition.* New York: American Elsevier Publishing Company, Inc., 1974.

[50] Miros, *TrueFace by Miros.* http://www.miros.com, 1998.

[51] Visionics, *Welcome to the Visionics Website.* http://www.visionics.com, 1998.

[52] Identicator, *Welcome to Identicator Technology!* http://www.identicator.com, 1998.

[53] American Biometric Company, *American Biometric Company Home Page.* http://www.biomouse.com, 1998.

[54] Veritel Corporation, *Veritel Corporation of America Homepage.* http://www.veritelcorp.com, 1998.

[55] T-Netix Inc., *T-Netix Home Page.* http://www.t-netix.com, 1998.

[56] ITT Industries, *SpeakerKey by ITT Industries.* http://www.speakerkey.com, 1998.

[57] J. G. Daugman and G. O. Williams, "A proposed standard for biometric decidability.," in *Proc. CardTech/SecureTech Conference*, (Atlanta, GA), pp. 223–234, 1996.

[58] S. G. Davies, "Touching big brother: How biometric technology will fuse flesh and machine," *Information Technology and People*, vol. 7, no. 4, pp. 60–69, 1994.